cadence™

Spectre® Circuit Simulator and Accelerated Parallel Simulator RF Analysis in ADE Explorer User Guide

Product Version 19.1
June 2020

© 1994–2020 Cadence Design Systems, Inc. All rights reserved.

Printed in the United States of America.

Cadence Design Systems, Inc. (Cadence), 2655 Seely Ave., San Jose, CA 95134, USA.

MMSIM contains technology licensed from, and copyrighted by: C. L. Lawson, R. J. Hanson, D. Kincaid, and F. T. Krogh Â© 1979, J. J. Dongarra, J. Du Croz, S. Hammarling, and R. J. Hanson Â© 1988, J. J. Dongarra, J. Du Croz, I. S. Duff, and S. Hammarling Â© 1990; University of Tennessee, Knoxville, TN and Oak Ridge National Laboratory, Oak Ridge, TN Â© 1992-1996; Brian Paul Â© 1999-2003; M. G. Johnson, Brisbane, Queensland, Australia Â© 1994; Kenneth S. Kundert and the University of California, 1111 Franklin St., Oakland, CA 94607-5200 Â© 1985-1988; Hewlett-Packard Company, 3000 Hanover Street, Palo Alto, CA 94304-1185 USA Â© 1994, Silicon Graphics Computer Systems, Inc., 1140 E. Arques Ave., Sunnyvale, CA 94085 Â© 1996-1997, Moscow Center for SPARC Technology, Moscow, Russia Â© 1997; Regents of the University of California, 1111 Franklin St., Oakland, CA 94607-5200 Â© 1990-1994, Sun Microsystems, Inc., 4150 Network Circle Santa Clara, CA 95054 USA Â© 1994-2000, Scriptics Corporation, and other parties Â© 1998-1999; Aladdin Enterprises, 35 Efal St., Kiryat Arye, Petach Tikva, Israel 49511 Â© 1999 and Jean-loup Gailly and Mark Adler Â© 1995-2005; RSA Security, Inc., 174 Middlesex Turnpike Bedford, MA 01730 Â© 2005

Open SystemC, Open SystemC Initiative, OSCI, SystemC, and SystemC Initiative are trademarks or registered trademarks of Open SystemC Initiative, Inc. in the United States and other countries and are used with permission.

Trademarks: Trademarks and service marks of Cadence Design Systems, Inc. contained in this document are attributed to Cadence with the appropriate symbol. For queries regarding Cadence's trademarks, contact the corporate legal department at the address shown above or call 800.862.4522. All other trademarks are the property of their respective holders.

Restricted Permission: This publication is protected by copyright law and international treaties and contains trade secrets and proprietary information owned by Cadence. Unauthorized reproduction or distribution of this publication, or any portion of it, may result in civil and criminal penalties. Except as specified in this permission statement, this publication may not be copied, reproduced, modified, published, uploaded, posted, transmitted, or distributed in any way, without prior written permission from Cadence. Unless otherwise agreed to by Cadence in writing, this statement grants Cadence customers permission to print one (1) hard copy of this publication subject to the following conditions:

1. The publication may be used only in accordance with a written agreement between Cadence and its customer.
2. The publication may not be modified in any way.
3. Any authorized copy of the publication or portion thereof must include all original copyright, trademark, and other proprietary notices and this permission statement.
4. The information contained in this document cannot be used in the development of like products or software, whether for internal or external use, and shall not be used for the benefit of any other party, whether or not for consideration.

Disclaimer: Information in this publication is subject to change without notice and does not represent a commitment on the part of Cadence. Except as may be explicitly set forth in such agreement, Cadence does not make, and expressly disclaims, any representations or warranties as to the completeness, accuracy or usefulness of the information contained in this document. Cadence does not warrant that use of such information will not infringe any third party rights, nor does Cadence assume any liability for damages or costs of any kind that may result from use of such information.

Restricted Rights: Use, duplication, or disclosure by the Government is subject to restrictions as set forth in FAR52.227-14 and DFAR252.227-7013 et seq. or its successor

Contents

Preface . . . 19
Using License Queuing . . . 21
Suspending and Resuming Licenses . . . 21
Related Documents for Spectre . . . 22
Third Party Tools . . . 23
Typographic and Syntax Conventions . . . 23
Introduction . . . 25
Simulation Basics . . . 25
Fundamentals of RF Simulation . . . 25
Shooting Newton and Harmonic Balance algorithm . . . 25
Shooting Method . . . 26
Harmonic Balance Method . . . 26
SpectreRF Analyses . . . 26
Periodic Analyses . . . 26
Quasi-Periodic Analyses . . . 28
Envelope Analysis . . . 30
APS and Parasitic Reduction . . . 30
Large vs. Small Signal Analysis . . . 32
Setting up a SpectreRF Simulation . . . 35
Using Virtuoso ADE Explorer . . . 35
A Testbench . . . 35
ADE Explorer . . . 36
Analysis form . . . 37
Direct Plot form . . . 37
Netlist Driven Flows . . . 38
Creating Multiple Tests in ADE Assembler . . . 39
Frequency Domain Analyses: Harmonic Balance . . . 53
Overview of Simulation Capabilities . . . 53
Large-Signal Harmonic Balance Overview . . . 53
Example . . . 55
Harmonic Balance Solves For Cosines . . . 63

© 1994-2020 All Rights Reserved.

Setting Harmonics Automatically 66
Setting Harmonics Manually 66
Oversample Factor 68
Trading off Harmonics and Oversample Factor 74
Two Input Frequencies 76
Frequency Cuts 77
Three Input Frequencies 81
Diamond Cut With Three Frequencies 82
Funnel Cut 83
Axis Cut 85
Multiple Frequency Dividers 85
Crossbox_hier 87
Crossbox 88
Widefunnel 89
Aliasing 89
Gain Compression Analysis 97
Large-Signal S-Parameters 99
Sine Representation of Square Wave 100
Convergence 101
Errpreset 101
Iteration Limit 102
Number of Non-Sinusoidal Sources 104
Harmonic Balance Starting Point 104
More Capabilities 105
Save and Recover 105
Writehb and Readhb 106
Sweeps and Restart 106
Itres 107
Freqdivide 107
Oscillators 107
Pinnode 108
Probe-Based Method 109
Oscillator Tuning Mode 109
Semi-Autonomous 110
Things to try to Achieve Convergence (Driven Circuits) 110
Things to try to Achieve Convergence (Oscillators) 111

© 1994-2020 All Rights Reserved.

Implementation in ADE Explorer . . . 112
Setting Frequencies, Harmonics, and Oversample . . . 112
Diamond Cut . . . 122
Funnel Cut . . . 123
Sweeps . . . 124
Freqdivide . . . 125
Compression Analysis . . . 126
Large-Signal S-Parameters . . . 131
Oscillator Additions . . . 134
Oscillator Tuning Mode . . . 137
Probe-Based Method . . . 140
Semi-Autonomous . . . 142
Commonly Used Harmonic Balance Options (Driven Circuits and Oscillators) . . . 147
Convergence tab . . . 148
Accuracy Tab . . . 151
Output Tab . . . 153
Reuse Tab . . . 154
Misc Tab . . . 162
Oscillator Options . . . 166
Key Outputs in the Spectre Window . . . 167
Examples . . . 176
ADE Explorer Setup for 1-tone Harmonic Balance (hb) Analysis . . . 179
Plotting Spectral Content . . . 184
Plotting dBm . . . 188
Sweeps . . . 191
Plotting Currents . . . 191
Compression Point . . . 195
Loadpull . . . 208
Power-Added Efficiency . . . 224
Oscillators . . . 226
Oscillator Tuning Mode Analysis . . . 234
Oscillator Tuning Mode With Sweeps . . . 238
Two Input Frequencies . . . 241
Multiple Divider Mode . . . 248
Compression Point . . . 250
IFFT . . . 253

© 1994-2020 All Rights Reserved.

Compression Vs Pout 255
Example Using the Harmonic Port with Harmonic Balance 258
Three Tone Harmonic Balance 268
Large-Signal IP3 272
Multi-Sinusoid hb Analysis 276
Low Memory Usage For hb Multi-Tone Analysis 284
Harmonic Balance AC Analysis (hbAC) 287
Small-signal Versus Large-Signal Analysis for Conversion Gain 287
Example: Conversion Gain (Down conversion) 288
Example: Conversion Gain (Up Conversion) 296
Overview of Simulation Capabilities 298
ADE Explorer Implementation 308
Circuit, Input port Setting, and ADE Explorer Setup for all the Examples in This Section 308
Using hbAC Analysis for Conversion Gain Measurement 311
Commonly Used HBAC Options 319
hbAC With Multiple hb Inputs 324
Rapid IP3/IP2 333
Triple Beat 338
Compression Distortion Summary 352
IM2 Distortion Summary 358
Modulated hbAC Analysis for an Oscillator 360
Sampled hbAC Analysis for a Mixer 367
Harmonic Balance Noise Analysis (hbnoise) 375
Small-Signal Versus Large-Signal Analysis for Noise 375
Overview of Simulation Capabilities 376
Example 377
Noise Output Near Zero Frequency 383
Noise Output Near The First Harmonic 385
Frequency Sweep 385
Maximum Sideband 386
Setting Harmonics and Sidebands 386
Noise Separation 386
Multiple hbnoise 387
Hbnoise after Multi-Tone hb 387
AM/PM Noise for Driven Circuits and Oscillators 387

© 1994-2020 All Rights Reserved.

Jitter . . . 391
Oscillators . . . 393
sampled(jitter) . . . 393
Noise Calculations in the Simulator . . . 393
Noise Factor and Noise Figure . . . 393
Input-Referred Noise . . . 396
Noise transfer function . . . 396
Commonly Used Options . . . 397
ADE Explorer Implementation . . . 399
Driven Circuits . . . 399
Oscillators . . . 425
Examples . . . 437
Driven Circuit Noise Setup . . . 437
Oscillators . . . 516
Creating an Oscillator Macro Source . . . 538
hbSP Analysis . . . 553
Overview of hbsp . . . 553
Example . . . 554
HBSP Options . . . 579
Harmonic Balance Transfer Function Analysis (HBXF) . . . 583
Example: Conversion Gain (Down Conversion) . . . 584
Overview of Simulation Capabilities . . . 592
Normal Conversion Gain Measurement (Specialized Analysis = None) . . . 593
Sampled HBXF . . . 603
ADE Implementation . . . 603
HBXF for a Normal Conversion Gain Measurement . . . 604
Sampled HBXF . . . 605
HBXF Options . . . 606
Examples for Driven Circuits . . . 609
HBXF Normal Conversion Gain Measurement . . . 609
Using HBXF Analysis for Conversion Gain Measurement . . . 610
Sampled HBXF Analysis . . . 618
Examples for Oscillators . . . 619
Conversion Gain From Power Supply to the Output Frequency . . . 619
Harmonic Balance Stability Analysis (HBSTB) . . . 625
Example: Variation in Loop Gain of an Opamp with Swept Input Signal Amplitude . 626

© 1994-2020 All Rights Reserved.

Overview of Simulation Capabilities . . . 634
ADE Implementation . . . 636
HBSTB Options . . . 638
Oscillator Example . . . 641
Single Input Large and Small-signal Analyses . . . 647
Overview of Simulation Capabilities . . . 647
Overview of Periodic Steady-State (pss) Analysis . . . 647
Example . . . 648
Harmonic Balance . . . 655
DC Principles . . . 660
Convergence . . . 662
Skipdc . . . 663
Transient Principles . . . 664
Convergence . . . 664
Relref . . . 664
Integration Methods . . . 669
Controlling Accuracy . . . 672
Shooting PSS . . . 675
Overview . . . 675
Oscillators . . . 676
Delay Time . . . 677
Piecewise Linear Sources for Power Supplies . . . 678
Tstab . . . 678
Beat Frequency . . . 678
Number of Harmonics . . . 679
Accuracy Defaults (errpreset) . . . 679
PSS Shooting Convergence . . . 681
ADE Implementation and Numerical Noise Floor . . . 682
Accuracy and Settings and Trade-Offs . . . 687
Which Engine Should be Selected? . . . 689
Important Outputs in the Spectre.out File . . . 690
Guidelines for Setting up Oscillators for Simulation . . . 694
Beat frequency . . . 694
Tstab . . . 694
Different ways to start the oscillator . . . 694
Saveinit- A way to save the startup waveform . . . 696

© 1994-2020 All Rights Reserved.

Oscillator Node- Used to estimate the frequency for the first pss iteration 696
ADE Implementation 697
Spectre Output File (Oscillators) 700
Dynamic Parameters in tstab 702
Oscillator Tuning Mode 703
PSS Options 705
Convergence tab 706
Accuracy Tab 709
Output Tab 724
Reuse Tab 732
Misc Tab 744
Help for Convergence Issues 749
Examples 752
Pss Shooting with a single input to an amplifier or the LO applied in a mixer 752
Plotting the output spectrum in dBm 759
Same Example Using Harmonic Balance 760
Loadpull 762
Two Input Frequencies 777
Sweeps 780
Compression Point 785
Voltage Gain 787
Power Gain 789
Power-Added Efficiency 792
Compression Versus Output Power 794
Power Gain Versus Output Power 795
Plotting THD 797
Large-signal IP3 799
Oscillator Simulation 807
Setting Dynamic Parameters in Tstab 807
Normal Oscillator Simulation 814
Plotting the Oscillator Startup Waveform 816
Plotting the Oscillator Output Spectrum 818
Additions for Oscillator Swept Tuning Voltage 819
Plotting Output Power Versus Tuning Voltage 822
Plotting Output Power Versus Output Frequency 823
Plotting the Output Frequency Versus the Tuning Voltage 826

© 1994-2020 All Rights Reserved.

Plotting the Modulation Sensitivity 829
Oscillator Tuning Mode Analysis 831
Oscillator Tuning Mode With Sweeps 835
Periodic AC Analysis (PAC) 840
Small-signal Versus Large-Signal Analysis for Conversion Gain 840
Example: Conversion Gain (Down Conversion) 841
Example: Conversion Gain (Up Conversion) 849
PAC General Principles 852
Overview of Simulation Capabilities 857
ADE Implementation 869
Circuit, Input port Setting, and ADE Explorer Setup for all the Examples in This Section 869
Using PAC Analysis for Conversion Gain Measurement 872
Sampled Circuits and Switched-capacitor Filters 881
Commonly Used PAC Options 882
PAC With Multiple PSS Inputs 886
Rapid IP3/IP2 894
Compression Distortion Summary 900
IM2 Distortion Summary 908
Modulated PAC Analysis 910
Sampled PAC Analysis 918
Periodic Noise Analysis (***Pnoise***) 926
Small-Signal Versus Large-Signal Analysis for Noise 926
Overview of Simulation Capabilities 927
Example 928
Principles of Pnoise 936
Noise Type=timeaverage 937
Noise Output Near The First Harmonic 938
Frequency Sweep 938
Maximum Sideband 940
Full-Spectrum Pnoise 940
Setting Harmonics and Sidebands 941
Noise Separation 941
Multiple Pnoise 942
AM and PM Noise for Driven Circuits and Oscillators 942
Noise Type=sampled(jitter) 946

© 1994-2020 All Rights Reserved.

Noise Calculations . . . 947
Phase Noise (Driven Circuit) . . . 947
Noise Factor and Noise Figure . . . 948
Input-Referred Noise . . . 951
Noise transfer function . . . 952
Commonly Used Pnoise Options . . . 953
ADE Implementation . . . 956
General Notes . . . 956
Driven Circuits . . . 957
Oscillators . . . 983
Examples . . . 997
Driven Circuit Noise Setup . . . 997
Multiple pnoise . . . 1036
Oscillators . . . 1076
Periodic Transfer Function Analysis (PXF) . . . 1117
Small-signal Versus Large-Signal Analysis for Conversion Gain . . . 1117
Example: Conversion Gain (Down conversion) . . . 1118
Example: Conversion Gain (Up Conversion) . . . 1127
Overview of Simulation Capabilities . . . 1129
Normal Conversion Gain Measurement (Specialized Analysis = None) . . . 1135
AM to PM measurement (Specialized Analysis = Modulated) . . . 1146
Sampled PXF . . . 1149
ADE Implementation . . . 1150
PXF for a Normal Conversion Gain Measurement . . . 1151
PXF Modulated (AM to PM Conversion) . . . 1153
PXF Sampled . . . 1155
PXF Options . . . 1159
Examples for Driven Circuits . . . 1167
PXF Normal Conversion Gain Measurement . . . 1167
Using PXF Analysis for Conversion Gain Measurement . . . 1169
PXF Modulated (AM to PM conversion) . . . 1179
Sampled PXF . . . 1188
Examples for Oscillators . . . 1199
Conversion Gain From Power Supply to the Output Frequency . . . 1199
Modulated PXF; Measuring AM to PM Conversion and Controlling Spurious Response 1204

© 1994-2020 All Rights Reserved.

Sampled PXF: Measuring Conversion Gain at a Threshold Crossing in a Ring Oscillator . . . 1216
Periodic Stability Analysis (PSTB) . . . 1228
Example: Variation in Loop Gain of an Opamp with Swept Input Signal Amplitude 1229
Overview of Simulation Capabilities . . . 1239
ADE Implementation . . . 1241
PSTB Setup . . . 1241
PSTB Options . . . 1243
The following options are not commonly used: . . . 1244
Oscillator Example . . . 1246
Periodic S-parameter Analysis (PSP) . . . 1252
Example: Down-Conversion Diode Mixer . . . 1252
Overview of Simulation Capabilities . . . 1270
ADE Implementation . . . 1272
PSP Setup . . . 1272
PSP Options . . . 1279
Examples . . . 1282
Swept Input Power In a PA . . . 1297
Multiple Input Large and Small-signal Analyses . . . 1311
Quasi-Periodic Steady-State Analysis (QPSS) . . . 1311
Example . . . 1312
QPSS Shooting Concepts . . . 1319
What Happens When you Start a QPSS Analysis? . . . 1327
ADE Implementation . . . 1331
QPSS Options . . . 1333
Example . . . 1346
Quasi-Periodic AC Analysis (QPAC) . . . 1386
Example . . . 1386
Qpac Concepts . . . 1393
ADE Implementation . . . 1403
Example . . . 1407
Quasi Periodic Noise Analysis . . . 1422
Overview . . . 1422
Example . . . 1423
Qpnoise Concepts . . . 1430
Measuring Noise Figure . . . 1432

© 1994-2020 All Rights Reserved.

Frequency Sweep 1432
Noise Separation 1435
Input-Referred Noise 1436
ADE Implementation 1437
Example 1443
Viewing Noise Separation Results 1462
QPXF Analysis 1494
Example: Conversion Gain 1495
Overview of Simulation Capabilities 1504
ADE Implementation 1512
QPXF for a Normal Conversion Gain Measurement 1512
QPXF Options 1514
Example 1517
Conversion Gain as a Function of Blocker Power 1531
QPSP Analysis 1539
Example: Down-Conversion Diode Mixer 1540
Overview of Simulation Capabilities 1559
ADE Implementation 1560
QPSP Setup 1560
QPSP Options 1570
Examples 1573
Envelope (ENVLP) Analysis 1587
Example 1587
Envelope Principles 1610
Envelope-Transient Simulation 1614
Transistor Level V/s Fast Envelope 1615
The Passband Model 1616
The Baseband Model 1617
Power Scaling in Baseband Mode 1617
WFreq 1618
When do I use Passband or Baseband? 1618
Fast Envelope Noise 1619
Wireless Mode in Envelope Analysis 1621
Supported Standards 1622
Wireless Analysis Vs. Traditional Envelope Following 1623
Limitation of Wireless Analysis 1623

© 1994-2020 All Rights Reserved.

Setting up the Envelope Analysis in ADE Explorer: Wireless Mode 1631
Setting up the Envelope Analysis in ADE Explorer: Using your own I and Q Files 1631
Frequency Modulated Input Signals 1655
Commonly Used Options 1661
Autonomous ENVLP Analysis (Oscillators) 1676
Examples 1693
Large-Signal S-Parameter Simulation (LSSP) 1695
Large-Signal S Parameters 1695
Large-Signal S parameters for a Two Port Circuit 1696
Circuit Setup 1697
ADE Explorer Setup 1701
Analysis Setup 1701
LSSP Amplitude Sweep 1704
Plotting the S11 Curve 1706
Plotting the Input Impedance Magnitude and Phase Curves 1710
AnalogLib Components Used in RF Simulation 1713
The Delayline Element 1713
The PORT Element 1715
Capabilities of the port Component 1715
Terminating the Port 1716
Parameters for the Port Component 1718
Port Parameters 1718
Using the Harmonic Port with Harmonic Balance 1719
Source type 1729
DC Parameters 1730
DC voltage 1730
Pulse Waveform Parameters 1731
PWL Waveform Parameters 1745
Waveform Entry Method 1747
Sinusoidal Waveform Parameters 1768
Modulation Parameters 1771
Display second sinusoid 1788
Display multi sinusoid 1790
Exponential Waveform Parameters 1795
Bit Waveform Parameters 1797

© 1994-2020 All Rights Reserved.

Bit Waveform Examples 1800
PRBS Waveform Parameters 1806
Jitter Generation 1825
PRBS Mode External Triggering 1845
PRBS Source with External Triggering Using a Pulse or PWL Signal 1848
Oscillator Macro Parameters 1851
Noise Parameters 1852
Small-Signal Parameters 1858
Temperature Effect Parameters 1860
Simulating Tabulated S-Parameters Using the Nport Component 1862
Convolution-Based Method 1869
Rational Fit Methods 1869
Using the nport Component 1871
Nport Compression 1877
Controlling Rational Fit Accuracy 1889
Using the relerr and abserr Parameters 1891
Using the rational order (ratorder) Parameter 1893
Troubleshooting 1893
Assessing the Quality of the Rational Interpolation 1894
Model Reuse 1899
Dcblock, dcfeed, indq, and capq 1902
Dcfeed 1902
Dcblock 1906
indq 1909
capq 1918
Reference: S-Parameter Equations 1925
Network Parameters 1926
Equations for Network Parameters 1926
Two-Port Scalar Quantities 1928
Equations for Two-Port Scalar Quantities 1929
Two-Port Gain Quantities 1931
Equations for Two-Port Gain Calculations 1932
Two-Port Network Circles 1934
Equations for Two-Port Network Circle 1935
Equation for VSWR (Voltage Standing Wave Ratio) 1938
Equation for GD (group delay) 1939

© 1994-2020 All Rights Reserved.

References . . . 1939
Cosimulation with MATLAB® and Simulink® . . . 1941
Introduction to Cosimulation with MATLAB . . . 1942
Software Requirements . . . 1942
Setting Up and Running a Cosimulation . . . 1942
Connecting the Coupler Block Into the System-Level Simulink Schematic . . . 1943
Determining How You Want to Start and Run the Cosimulation . . . 1947
Generating a Netlist for the Lower-Level Block . . . 1948
Preparing the Netlist When Using ADE Explorer . . . 1948
Preparing the Netlist Without Using a Graphical User Interface . . . 1955
Running the Cosimulation . . . 1957
Starting the Two Applications Separately . . . 1957
Starting SpectreRF Manually and MATLAB Automatically . . . 1958
Starting MATLAB Manually and SpectreRF Automatically . . . 1958
MATLAB Support Matrix . . . 1959
Top-down RF Design Methodology . . . 1961
Top-Down Design of RF Systems . . . 1962
Use Model for Top Down Design . . . 1963
Baseband Modeling . . . 1966
Example Comparing Baseband and Passband Models . . . 1968
rfLib Library Overview . . . 1980
Use Model and Design Example . . . 1984
Opening a New Schematic Window . . . 1985
Opening the Analog Environment . . . 1986
Constructing the Baseband Model for the Receiver . . . 1986
Setting Variable Values for the Receiver Schematic . . . 2024
Setting Up and Running a Transient Analysis . . . 2027
Examining the Results: Eye Diagram, Histogram, and Scatter Plot . . . 2029
Computing Minimized RMS Noise Using the Optimizer . . . 2041
Summarizing the Design Procedure . . . 2058
Creating a Passband View of the Architectural Model . . . 2058
Comparing Baseband and Passband Models . . . 2062
Relationship Between Baseband and Passband Noise . . . 2065
Introduction to Analysis . . . 2066
Preparation Steps for Analyses . . . 2067

© 1994-2020 All Rights Reserved.

Documents That Ship in the Software Hierarchy 2075
SpectreRF_simulink_example.pdf 2076
EnvelopeAN.pdf 2077
LSSP_AN.pdf 2077
MatlabWorkshop.pdf 2077
PLL_Jitter_AN.pdf 2078
NS_AN.pdf 2078
PSRR_Drv_AN.pdf 2079
readme.txt 2079
HB_AN.pdf 2079
LTJM_AN.pdf 2080
PerturbationAN.pdf 2080
PSRR_Osc_AN.pdf 2081
RF_Blocks_AN.pdf 2081
JitterAN.pdf 2082
MatlabAN.pdf 2082
PstbAN.pdf 2083

Index 2085

© 1994-2020 All Rights Reserved.

© 1994-2020 All Rights Reserved.

Preface

Spectre® circuit simulator or Spectre® Accelerated Parallel Simulator (Spectre APS) base products with the Spectre RF analysis option provide simulation capabilities for RFIC designers. The simulators:

- Support efficient calculation of the operating point, transfer function, noise, and signal distortion of common RF and communication circuits, such as mixers, LNA, oscillators, sample and holds, and switched capacitor filters.
- Support a multi-technology simulation (MTS) mode that enables the simulation of systems consisting of blocks designed with different processes, such as RF System-in-Package (SIP).

This user guide assumes that you are familiar with:

- RF circuit design
- SPICE simulation
- Virtuoso® Analog Design Environment (ADE)

The Spectre® circuit simulator, the high performance Spectre® APS, and the Spectre RF analysis options are part of the MMSIM (multi-mode simulation) portfolio and are accessible using individual a la carte licenses or MMSIM tokens. Note that mixing of tokens and a la carte license is not allowed. Table -1 on page 19 and Table -2 on page 20 show the capabilities offered with the base products Spectre circuit simulator or Spectre APS and the Spectre RF analysis option.

Table -1 Spectre® Simulator with Spectre RF Analysis Option

Features
Transient Noise Analysis
DC, small-signal analyses and transient
Monte Carlo, DC mismatch, Parametric sweep
RF Harmonic Balance

© 1994-2020 All Rights Reserved.

Features
RF Shooting Newton
Large-signal Analysis - Periodic and Quasi-periodic (HB, PSS and QPSS)
Noise Analysis - Periodic and Quasi-periodic (pnoise, QPNoise, sampled, jitter)
Small-signal Analysis - Periodic and Quasi-periodic (PAC, PXF, PSP, QPAC, QPXF, QPSP)
Periodic Stability Analysis (PSTB)
RF Transmission Line Library
Envelope Analysis (AM, PM, FM, Autonomous)
Wireless Analysis
Wireless Standard Sources (such as LTE, Zigbee, and 802.11n)
Rapid Distortion Analyses - Perturbation-based IP2 and IP3
Co-simulation with Simulink® from The MathWorks
MMSIM Toolbox for MATLAB® from The MathWorks

Table -2 Spectre® Accelerated Parallel Simulator with Spectre RF Analysis Option

Features
All analysis and features in Spectre
RF Harmonic Balance
RF Shooting Newton
Large-signal Analysis - Periodic and Quasi-periodic (HB, PSS and QPSS)
Noise Analysis - Periodic and Quasi-periodic (pnoise, QPNoise, sampled, jitter)
Small-signal Analysis - Periodic and Quasi-periodic (PAC, PXF, PSP, QPAC, QPXF, QPSP)
Periodic Stability Analysis (PSTB)
RF Transmission Line Library
Envelope Analysis (AM, PM, FM, Autonomous)
Wireless Analysis
Wireless Standard Sources (such as LTE, Zigbee, and 802.11n)

© 1994-2020 All Rights Reserved.

Features
Rapid Distortion Analyses - Perturbation-based IP2 and IP3

Cadence offers multi-core simulation up to 64 cores for Spectre RF analysis, enabled with the base product Spectre APS along with Spectre RF analysis and the Spectre CPU Accelerator options.

Please contact your account representative for more details on the licensing and packaging.

Using License Queuing

You can turn on license queuing by using the `lqtimeout` command line option:

```
spectre +lqtimeout time_in_seconds
```

If a license is not available when you begin a simulation job, the Spectre circuit simulator and Spectre APS wait in queue for a license for the specified time. If you specify the value 0 for this option, the Spectre circuit simulator waits indefinitely for a license. The `lqtimeout` option ha s no default value for the standalone Spectre circuit simulator. If you invoke Spectre through the Analog Design Environment, the default value for `lqtimeout` is 900 seconds.

You can use the `lqsleep` option to specify the interval (in seconds) at which the Spectre circuit simulator should check for license availability. The default value for `lqsleep` is 30 seconds.

```
spectre +lqsleep interval
```

For more information on any of the above options, see `spectre -h`.

Suspending and Resuming Licenses

You can direct Spectre and Spectre APS to release licenses when suspending a simulation job. This feature is aimed for users of simulation farms, where the licenses in use by a group of lower priority jobs may be needed for a group of higher priority jobs. To enable this feature, simply start Spectre with the `+lsuspend` command line option. In the Solaris environment, press `ctrl+z` to suspend the Spectre license. All licenses are checked in. To resume simulation, press `fg`. These keystrokes may not work if you have changed the default key bindings.

For information on tracking token licensing, see the *Virtuoso® Software Licensing and Configuration Guide*.

© 1994-2020 All Rights Reserved.

In Virtuoso® Analog Design Environment, the `lqtimeout` and `lqsleep` options are controlled by the following options:

```
spectre.envOpts lsuspend boolean t
spectre.envOpts licQueueTimeOut string "900"
spectre.envOpts licQueueSleep string "30"
```

Related Documents for Spectre

This user guide contains information about the functionality. The following documents provide more information about the Spectre RF analysis option with Spectre, Spectre APS, and related products.

- The Spectre circuit simulator is often run within the analog circuit design environment, under the Cadence design framework II. To see how the Spectre circuit simulator is run under the analog circuit design environment, read the *Virtuoso Analog Design Environment User Guide.*
- To learn more about specific parameters of components and analyses, consult the Spectre online help (`spectre -h`).
- To learn more about the equations used in the Spectre circuit simulator, consult the *Spectre Circuit Simulator Components and Device Models Manual*.
- The Spectre circuit simulator also includes a waveform display tool, Virtuoso Visualization and Analysis tool, to use to display simulation results. For more information about the tool, see the *Virtuoso Visualization and Analysis User Guide*.
- For more information about using the Spectre circuit simulator with Verilog-A, see the *Verilog-A Language Reference* manual.
- For more information about RF theory, see *Spectre Circuit Simulator RF Analysis Theory*.
- For more information about how you work with the design framework II interface, see *Cadence Design Framework II Help*.
- For more information about specific applications of Spectre analyses, see *The Designer's Guide to SPICE & Spectre*[1].

1. Kundert, Kenneth S. *The Designer's Guide to SPICE & Spectre.* Boston: Kluwer Academic Publishers, 1995.

© 1994-2020 All Rights Reserved.

Third Party Tools

To view any `.swf` multimedia files, you need:

- Flash-enabled web browser, for example, Internet Explorer 5.0 or later, Netscape 6.0 or later, or Mozilla Firefox 1.6 or later. Alternatively, you can download Flash Player (version 6.0 or later) directly from the Adobe website.
- Speakers and a sound card installed on your computer for videos with audio.

Typographic and Syntax Conventions

This list describes the syntax conventions used for the Spectre circuit simulator.

`literal`	Nonitalic words indicate keywords that you must enter literally. These keywords represent command (function, routine) or option names, filenames and paths, and any other sort of type-in commands.
`argument`	Words in italics indicate user-defined arguments for which you must substitute a name or a value. (The characters before the underscore (_) in the word indicate the data types that this argument can take. Names are case sensitive.
`|`	Vertical bars (OR-bars) separate possible choices for a single argument. They take precedence over any other character.
`[ ]`	Brackets denote optional arguments. When used with OR-bars, they enclose a list of choices. You can choose one argument from the list.
`{ }`	Braces are used with OR-bars and enclose a list of choices. You must choose one argument from the list.

© 1994-2020 All Rights Reserved.

. . .	Three dots (...) indicate that you can repeat the previous argument. If you use them with brackets, you can specify zero or more arguments. If they are used without brackets, you must specify at least one argument, but you can specify more.

The language requires many characters not included in the preceding list. You must enter required characters exactly as shown.

© 1994-2020 All Rights Reserved.

1

Introduction

Simulation Basics

Fundamentally, SpectreRF provides large-signal and small-signal analyses. Large-signal analyses PSS, QPSS, HB, and Envelope solve all the equations, and capture all the harmonics that are produced by the system. Large-signal effects like compression are taken into account. Because all the nonlinearities of the system need to be solved for, the simulation time is longer than the simulation time for small-signal analyses where the large-signal effects like compression are not solved for. The nonlinearity solved for in the large-signal analysis is taken into account, so frequency translation effects caused by large signals are present in the small-signal analyses. Therefore, conversion gain and noise are accurate for systems that translate frequencies.

Fundamentals of RF Simulation

SpectreRF provides solutions for single or multi tone periodic systems using periodic steady-state (PSS) or quasi-periodic steady state (QPSS), and the small-signal analyses provided for all three. Noise simulations that include frequency translation are provided using Pnoise, hbnoise, or QPnoise. To measure small-signal conversion gain, pac, QPAC, pxf, and QPXF analyses are provided. To measure IP3, rapid IP3 is provided. To measure power gain, PSP and QPSP are provided. This is an extension of S-Parameters for systems that translate frequency. Non-periodic systems can be simulated using envelope.

Shooting Newton and Harmonic Balance algorithm

SpectreRF provides a choice of simulation engines between the traditional *Shooting Newton method* (or simply, *Shooting method*) and the *Harmonic Balance method* (HB) with most analyses. The Harmonic Balance engine complements the capabilities of the Shooting method.

The combination of a PSS or QPSS analysis using the Shooting method with a time-varying small-signal analyses is efficient for circuits that respond in a strongly nonlinear manner to

© 1994-2020 All Rights Reserved.

the LO or the clock. Consequently, you can use the SpectreRF simulations with the Shooting method to simulate strongly nonlinear circuits, such as switched-capacitor filters, switching mixers, chopper-stabilized amplifiers, PLL-based frequency multipliers, and sample-and-holds. You can use the SpectreRF simulations with the Harmonic Balance method to simulate circuits that produce a reasonably small number of harmonics, such as Low-Noise Amplifiers (LNAs) or Gilbert cell mixers.

Generally, PSS is used for circuits with a single input frequency. Either Shooting or Harmonic Balance can be used. Usually QPSS or HB is used for circuits that have more than one input frequency. QPSS runs much faster and requires less memory when Harmonic Balance is used as the engine. Use Shooting only for very nonlinear circuits like samplers with the sample clock and one or more inputs applied, or switched-capacitor circuits with the switch clock and one or more inputs applied.

Shooting Method

SpectreRF has traditionally used an engine known as the *Shooting method* [kundert90] to implement periodic and quasi-periodic analyses and the envelope analysis. The Shooting method is a time domain method.

Harmonic Balance Method

The Harmonic Balance engine supports frequency domain Harmonic Balance analyses. It provides efficient and robust simulation for circuits that produce a reasonable small number of harmonics. The Harmonic Balance engine is supported on the Solaris, Linux, HP and IBM platforms for both 32 and 64 bit architectures. See *Spectre Circuit Simulator RF Analysis Theory* for more information on the Harmonic Balance engine.

SpectreRF Analyses

The Spectre® circuit simulator RF analysis (SpectreRF) adds capabilities to the Spectre circuit simulator, such as direct, efficient computation of steady-state solutions and simulation of circuits that translate frequency. You use the SpectreRF analyses in combination with the Fourier analysis capability of the Spectre circuit simulator and with the Verilog®-A behavioral modeling language.

Periodic Analyses

SpectreRF adds periodic large and small-signal analyses to Spectre simulation.

© 1994-2020 All Rights Reserved.

- **Periodic Steady-State Analysis, PSS (Large-Signal)**

  PSS captures the full non-linear behavior of the circuit usually with a single input applied to the circuit. Shooting or Harmonic Balance can be used for PSS.

- **Periodic AC Analysis, PAC (Small-Signal)**

  PAC allows the small-signal conversion gain and allows the small-signal IP3 to be measured.

- **Periodic S-parameter Analysis, PSP (Small-Signal)**

  PSP allows the small-signal power conversion gain to be measured, and allows the input match at (for example) the RF and the output match at the IF to be measured in the same run.

- **Periodic Transfer Function Analysis, PXF (Small-Signal)**

  PXF can be used to measure small-signal conversion gain to be measured and also PSRR with frequency translations can be measured.

- **Periodic Noise Analysis, Pnoise (Small-Signal)**

  Pnoise allows the noise of systems that translate frequencies to be measured accurately.

- **Periodic Stability Pstb** (**Small-Signal**)

  Pstb measures the open-loop gain and phase for systems where there is a large signal applied. The large signal can cause nonlinearity which might affect the stability of feedback systems.

- **Harmonic balance analysis, HB**

  HB calculates the harmonics and mixing products for all the inputs.

- **Harmonic Balance AC analysis, HBAC**

  HBAC allows the small-signal conversion gain and the small-signal IP3 to be calculated.

- **Harmonic Balance Noise, HBnoise**

  HBnoise allows the Pnoise of systems that translate frequencies to be calculated accurately. Hbxf is provided for the measurement of conversion gain from the sources in the circuit to the output specified for the noise analysis.

- **Harmonic Balance S-Parameter, Hbsp (Small-Signal)**

  Hbsp allows the small-signal power conversion gain to be measured, and allows the input match at (for example) the RF and the output match at the IF to be measured in the same run.

© 1994-2020 All Rights Reserved.

For details on the periodic analyses, see *Spectre Circuit Simulator RF Analysis Theory*.

Periodic Steady-State (PSS) analysis is a large-signal analysis that directly computes the periodic steady-state response of a circuit. With PSS, simulation times are independent of the time constants of the circuit, so PSS can quickly compute the steady-state response of circuits with long time constants, such as high-Q filters and oscillators. You can also sweep frequency or other variables using PSS.

Generally, PSS is used for circuits that have a single input frequency. Although it can simulate circuits with multiple inputs as long as fewer than 50 periods of the highest input frequency or less are simulated, it solves for all of the harmonics produced by the system. If you have a system that has 900MHz and 1GHz, and you want the harmonics though 3.5GHz, this requires 35 harmonics of 100MHz, and is a reasonable problem for PSS to simulate. If the system had 1GHz and 1GHz plus 1MHz, and you wanted harmonics through 3.5GHz, PSS would need to solve for 3,500 harmonics of 1MHz, which would take a very long time, and lots of computer memory. In this case, use QPSS or HB because these analyses only solve for the actual mixing products that are produced by the system. This is a much smaller number of harmonics, thus the runtime is much faster, and it is just as accurate.

After completing a PSS analysis, the SpectreRF simulator can model frequency conversion effects by performing one or more of the periodic small-signal analyses, Periodic AC analysis (PAC) (caused by the nonlinearity from the large signal applied to the circuit), Periodic S-parameter analysis (PSP), Periodic Transfer Function analysis (PXF), Periodic Stability (Pstb), and Periodic Noise analysis (Pnoise). The periodic small-signal analyses are similar to the Spectre L AC, SP, XF, STB, and Noise analyses, but you can apply the periodic small-signal analyses to periodically driven circuits that exhibit frequency conversion. Examples of important frequency conversion effects include conversion gain in mixers, noise in oscillators, and filtering using switched-capacitors.

Therefore, with periodic small-signal analyses you apply a small signal at a frequency that may be non commensurate (not harmonically related) to the small signal fundamental. This small signal is assumed to be small enough so that it is not distorted by the circuit.

Quasi-Periodic Analyses

SpectreRF adds quasi-periodic large and small-signal analyses to Spectre L simulation. Quasi-periodic means that even systems with input signals that don't have a rational number as the fundamental periodicity of the system can still be simulated. If for example, if a system has an input at 1GHz and an input which is a multiple of pi, this system can still be simulated with QPSS.

- **Quasi-Periodic Steady-State Analysis, QPSS (Large-Signal)**

© 1994-2020 All Rights Reserved.

QPSS captures the full non-linear behavior of the circuit with two or more inputs applied to the circuit. For most systems, Harmonic Balance is more efficient.

- **Quasi-Periodic AC Analysis, QPAC (Small-Signal)**

  QPAC allows the measurement of small-signal conversion gain. QPAC takes into account all the harmonics that are calculated by the QPSS analysis.

- **Quasi-Periodic S-parameter Analysis, QPSP (Small-Signal)**

  QPSP allows the small-signal power conversion gain to be measured, and allows the input match at (for example) the RF and the output match at the IF to be measured in the same run. It also allows the measurement of stability factor on amplifiers with large signals present at the input.

- **Quasi-Periodic Transfer Function Analysis, QPXF (Small-Signal)**

  QPXF can be used to measure small-signal conversion gain and also PSRR with frequency translations taken into account can be measured.

- **Quasi-Periodic Noise Analysis, QPnoise (Small-Signal)**

  QPnoise allows the noise of systems that translate frequencies to be measured accurately. All the harmonics of the tones applied in the QPSS are taken into account. Therefore, QPnoise is useful for measuring receiver desensitization when a blocker is applied at the RF input.

- **Quasi-Periodic S-parameter Analysis, QPSP (Small-signal)**

  QPSP allows the measurement of S-Parameters for circuits that have two or more inputs and that translate frequency. For example, S11 is at the input frequency, S22 is at the output frequency, and S21 measures the forward conversion gain.

For details on the quasi-periodic analyses, see *Spectre Circuit Simulator RF Analysis Theory*.

Quasi-Periodic Steady-State (QPSS) analysis, a large-signal analysis, is used for circuits with multiple large tones. With QPSS, you can model periodic distortion and include harmonic effects.

QPSS computes both a large signal, the periodic steady-state response of the circuit, and also the distortion effects of a specified number of moderate signals, including the distortion effects of the number of harmonics that you choose. In Shooting, one signal must be designated large. This signal should be the signal that causes the largest amount of distortion in the system, or the signal that is non-sinusoidal. In Harmonic Balance, tstab should be set to yes for this signal. With QPSS, you can apply one or more additional signals at frequencies not harmonically related to the large signal, and these signals can be large enough to create

© 1994-2020 All Rights Reserved.

distortion. Shooting requires that these signals be sinusoidal, but Harmonic Balance allows one additional signal to be a pulse signal.

Quasi-Periodic Noise (QPnoise) analysis is similar to a noise analysis, except that it includes frequency conversion and intermodulation effects. QPnoise analysis is useful for predicting the noise behavior of mixers, switched-capacitor filters and other periodically or quasi-periodically driven circuits. Quasi-periodic systems are systems that have an irrational number as one of the input frequencies. Because of this, there is no exact period of the output waveform.

The Quasi-Periodic AC (QPAC), Quasi-Periodic S-parameter (QPSP) and Quasi-Periodic Transfer Function (QPXF) analyses all work in a similar way as the Spectre L AC, SP and XF analyses except that they include the frequency translations that occur because of a large-signal LO or clock input.

Envelope Analysis

Envelope analysis allows RF circuit designers to efficiently and accurately predict the envelope response of the RF circuits used in communication systems. This allows the ACPR and EVM to be calculated for power amplifier and transmit modulators to be calculated. The new fast envelope capability in release 10.1 can run 100 times faster than traditional envelope with very little loss in accuracy.

For details on the Envelope analysis, see *Spectre Circuit Simulator RF Analysis Theory.*

APS and Parasitic Reduction

Parasitic reduction is used in post parasitic re-simulations. It is RC reduction only with L and K values are preserved. Reducing the RC networks reduces the number of nodes in the circuit, thereby reducing the number of matrix calculations, so the simulation runs faster. It takes multiple poles at high frequencies and reduces them to a smaller order RC network with lower pole frequencies. When RC reduction is selected with the default settings, the minimum pole frequency in the reduced netlist is 1GHz. This might cause errors for RF circuits because the pole might be below the operating frequency. When RF is selected, the minimum pole frequency is raised to 30GHz. This will result is less reduction, and thus longer simulation times compared to the default setting. Selecting Fmax allows the user to set the minimum pole frequency in GHz. If you want 10GHz as the minimum pole frequency set the `Fmax`(GHz) field to 10, and not 10G. This allows you to make trade-offs between accuracy and runtime for your needs.

APS makes the CMOS models more efficient and allowing multi-threading for the device current evaluation. This will be relatively more efficient on schematic-level circuits and adds

© 1994-2020 All Rights Reserved.

the transient APS algorithms for DC and tstab solutions. It also allows multi-threading for all the matrix operations for Harmonic Balance and shooting and their small-signal analyses.

The processor affinity field should be used when you are running two or more jobs on the same multi-core machine. For example, if you have an 8 core machine and you are running two 4-thread jobs; assigning one job to cores 0 through 3 and the other to 4 though 7 saves time by having the operating system always use the same range of cores. This can save some time since the OS has less work to do. Note that if one of the cores you assign the job to already has a job, the entire simulation will run slowly because that one core will need to run two jobs by sharing time between them. The assumption is that all the threads will run on cores that are fully dedicated to the APS job.

© 1994-2020
All Rights Reserved.

Large vs. Small Signal Analysis

SpectreRF provides a variety of time-varying small signal analysis for both periodic and quasi-periodic circuits. These small-signal analyses accurately model the frequency translation effects of time-varying circuits. Rather than using traditional small-signal analyses for circuits that exhibit no frequency translation, such as amplifiers and filters, you can simulate circuits that translate frequencies using time-varying small-signal analyses.

Circuits designed to translate from one frequency to another include mixers, samplers, frequency multipliers, phase-locked loops and parametric oscillators. Such circuits are commonly found in wireless communication systems.

Other circuits that translate energy between frequencies as a side effect include oscillators, switched-capacitor and switched-current filters, chopper-stabilized and parametric amplifiers, and sample-and-hold circuits. These circuits are found in both analog and RF circuits.

The quasi-periodic small-signal analyses accurately model the small signal characteristics of circuits with a quasi-periodic operating point, such as mixers with multiple LO frequencies or large RF inputs. The periodic small-signal analyses are more useful for circuits with a single input frequency.

Applying a time-varying small-signal analysis is a two-step process.

First, the simulator ignores the small input or noise signals while performing a PSS, QPSS or Hb analysis to compute the steady-state response to the remaining large-signals, such as the LO or the clock. The PSS or QPSS analysis is the full nonlinear solution for the circuit.

For each subsequent small-signal analysis, the simulator uses the nonlinear operating point computed by the PSS, HB, or QPSS analysis to predict the circuit response to a small sinusoid at an arbitrary frequency. You can perform any number of small-signal analyses after calculating the time-varying large-signal operating point.

The input signals for the small-signal analyses must be sufficiently small that the circuit does not respond to them in a significantly nonlinear fashion. You should use input signals that are at least 10 dB smaller than the 1 dB compression point. This restriction does not apply to the signals you apply in the large-signal analysis.

This two-step process is widely applicable because most circuits that translate frequency react in a strongly nonlinear manner to one stimulus, usually either the LO or the clock, while they react in a nearly-linear manner to other stimuli such as the inputs. A mixer is a typical example. Its noise and conversion characteristics improve if it is discontinuously switched between two states by the LO, yet it must respond linearly to the input signal over a wide dynamic range.

© 1994-2020 All Rights Reserved.

To analyze a mixer with a small RF input and a single LO, you should use a PSS large-signal analysis using shooting or hb as the engine followed by one or more of the PAC, Pnoise, PSP or PXF small-signal analyses.

If the mixer has a small RF input and a large blocker as well as the LO, then a QPSS analysis would be the more appropriate large-signal analysis. Follow the QPSS analysis with one or more of the QPAC, QPNoise, QPSP or QPXF small-signal analyses for the RF input. The basic idea is this: If you have signals in your circuit that cause large-signal effects (like a blocker) this signal must be applied in a large-signal analysis like PSS for one input, or QPSS (or HB) for multiple inputs. For signals that are largely linear (like the conversion product of a receive mixer), you can use small-signal analyses which run faster. If the signals are largely sinusoidal, then hb and the small-signal analyses hbac, hbnoise, and hbsp can also be used.

Some circuits, such as frequency dividers, generate subharmonics. PSS can simulate the large-signal behavior of such circuits if you specify the period T to be that of the sub-harmonic. For other circuits, such as delta-sigma modulators, the periodically driven circuits respond chaotically, and you must use transient analysis rather than the PSS or QPSS analyses.

With the time-varying small-signal analyses such as QPAC or PXF, unlike traditional small-signal analyses such as AC or XF, there are many transfer functions between any single input and output due to mixing with harmonics. Usually, however, only one or two harmonics provide useful information. For example, when you analyze the down-conversion mixers found in receivers, you want to know about the transfer function that maps the input signal at the RF to the output signal at the IF, which is usually the LO minus the RF.

© 1994-2020 All Rights Reserved.

© 1994-2020 All Rights Reserved.

2

Setting up a SpectreRF Simulation

Using Virtuoso ADE Explorer

Starting with IC6.1.7, Cadence offers a new Virtuoso ADE product suite that enables you to fully explore, analyze, and verify a design against design goals so that the design intent can be maintained throughout the design cycle. The new generation of ADE product suite includes ADE Explorer, ADE Assembler, and ADE Verifier products.

You can use the tools in the new Virtuoso ADE product suite as your simulation environment because of the benefits they provide. The netlister netlists directly from the schematic. The Direct Plot Form automates all of the standard RF measurements. Cross-probing to the schematic is provided. Easy set-up of the analyses to be run is provided. ADE Explorer runs a single test and allows parallel processing of sweeps, corners and Monte Carlo. In ADE Assembler, multiple tests can be run at the same time. In ADE Verifier, you can import simulation data and measurements against the specs from all the members of the design team to get a real-time update of the design progress.

For more information on the new products, refer to the *Virtuoso ADE Explorer User Guide*, *Virtuoso ADE Assembler User Guide*, and *Virtuoso ADE Verifier User Guide*.

A Testbench

A testbench is the schematic with the sources set up to provide the stimulus, and the different analyses to provide the measurements. In the schematic, make sure that the sources used for inputs and bias are at the top level, and have the circuit instantiated as a symbol (which could have levels of hierarchy below that). This allows these sources to be connected to the circuit but not have any connection to layout.

The new ADE tools create a maestro view (see Creating Multiple Tests in ADE Assembler) that stores the setups of all the different tests. The different tests are like ADE L states.

Note: The products in the new ADE product suite can load ADE states created in ADE L and ADE XL, however, they do not have the ability to save ADE states in the same manner as ADE L. You can create multiple tests in ADE Assembler and run these tests individually in ADE

© 1994-2020 All Rights Reserved.

Explorer, when required. Refer to Creating Multiple Tests in ADE Assembler on page 39 to learn how to create multiple tests in ADE Assembler.

ADE Explorer

ADE Explorer, incorporates some useful features of ADE XL while maintaining the simplicity and ease of use of ADE L. ADE Explorer is like ADE L in that it can run a single design and simulation setup at a time. It adds the ability to run parallel simulations in sweeps. Corners and Monte Carlo are also available along with real-time tuning (RTT) which is useful for rapidly tuning filters or matching networks. ADE Explorer also allows very easy transitions to ADE Assembler, which allows multiple tests and is similar to ADE XL. Like ADE XL, the simulation environment is in one tab, and the schematic is in another in ADE Explorer, as shown highlighted in the figure below.

You need to specify the following essential settings in the Virtuoso® ADE Explorer Setup before running a simulation — design variables, temperature, simulator, analysis, model files,

© 1994-2020 All Rights Reserved.

and other setup files. You can also define the various outputs that are to be saved or plotted during a simulation run.

When you run a simulation, the results are saved in a predefined formats and can be used by other tools such as the Virtuoso Visualization and Analysis XL for further analysis. If there are signals in the simulation results, they are by default plotted in the Virtuoso Visualization and Analysis XL Graph window.

Analysis form

The *Choosing Analysis* form is provided for each analysis type you want to run. The *Choosing Analysis* form lists the sources in your circuit at the top. The number of harmonics and oversample factor are set in the edit fields just below the list. Accuracy is set by selecting liberal, moderate, or conservative. In general, to get a quick idea, choose liberal. For most simulations, choose moderate. When a more accurate simulation is desired, select conservative. If you have an oscillator, select the oscillator button. Sweeps can be set up in the form directly, or by using the parametric plot tool in ADE.

Each analysis has its own *Choosing Analysis* form where the appropriate information is entered for that analysis.

Direct Plot form

The *Direct Plot* form automates the plotting of signals. At the top, all the different analyses that were run are listed. Just below are the different things that can be plotted. In SpectreRF, both the time and frequency domain results are provided regardless of whether the simulation is done in the time domain using Shooting, or in the frequency domain using Harmonic Balance. Peak or RMS results are available, and when frequency domain results are present, modifiers are allowed so that dB or phase could be selected for the results.

© 1994-2020 All Rights Reserved.

When a selection is made in the schematic, the waveform tool displays the result. In ADE Explorer, by default, the waveform is docked in the ADE Explorer window, as shown in the figure below.

You can click the Launch ViVA icon (shown highlighted above) to display the waveform separately in the Virtuoso Visualization and Analysis tool (ViVA).

Netlist Driven Flows

Netlist-driven flows start with a netlist. This might be done manually, or from a netlister that is developed in house. The netlister may or may not add the analysis statements to the netlist. One advantage to running from the netlist is the ability to alter a value and then run again, or to run multiple analyses of the same type with different settings.

© 1994-2020 All Rights Reserved.

To get spectre help for the netlist-driven flow, type `spectre -h` in a shell window. This gives a list of topics that have help available for, and then you might for example type `spectre -h pss`. When you do this, you will see the analysis statement and all the options along with all the values that are available.

When the netlist has the proper analysis statements, you can run spectre. Examples of this are shown below:

```
spectre netlist.scs
```

```
spectre +aps +mt=2 +log spectre.out netlist.scs (This runs APS with 2 threads and
     a log file spectre.out)
```

When Spectre runs, the results are saved in the <*netlist_name*>.raw directory. To view the results, type `viva&` in a shell window, and then plot the waveforms from the Results Browser and the Calculator.

Creating Multiple Tests in ADE Assembler

The tools in the new ADE product suite do not have the capability to save ADE states. However, you can create and use the equivalent of ADE states in the new ADE environment. You can create multiple tests in ADE assembler that correspond to multiple ADE states. These tests are saved in the maestro view.

Note: You have to create the maestro view when any of the new ADE tools is first started. When you run the simulation, or when you specifically save the setup, the maestro view is updated.

The following is an example of creating multiple tests in the ADE Assembler:

© 1994-2020 All Rights Reserved.

1. Open a design in ADE Explorer. To do this, launch virtuoso and select *Tools - Library Manager* from the Virtuoso window. The *Library Manager* window is displayed, as shown below.

2. Select a library from the *Library* section, cell from the *Cell* section, and double-click on the *schematic* view in the *View* section.

© 1994-2020
All Rights Reserved.

3. The schematic is displayed in the *Virtuoso Schematic Editor* window, as shown below.

4. Select *Launch - ADE Explorer* to open the ADE Explorer window.

© 1994-2020
All Rights Reserved.

The Launch ADE Explorer form is displayed, as shown below.

5. Select the *Create New View* option and click *OK*.

   The *Create new ADE Explorer view* window is displayed, as shown below.

6. Click OK.

© 1994-2020
All Rights Reserved.

ADE Explorer will start in a new tab, as shown below.

7. Click on the test *ExampleLibRF:InaSimple:1* twice and change the name to *Setup*.
8. Press *Enter*.

© 1994-2020 All Rights Reserved.

The name of the test will change to *Setup*, as shown below.

9. Select *Analyses - Choose*

© 1994-2020 All Rights Reserved.

The *Choosing Analyses* form is displayed, as shown below.

Note: You can also open the *Choosing Analysis* form by expanding the *Analyses* option in the *Setup* assistant of the ADE Explorer window and clicking the *Click to add analysis* text.

© 1994-2020
All Rights Reserved.

10. In the *Choosing Analyses* form, select the *dc* option from the *Analysis* section.

11. Select the *Save DC Operating Point* option.

12. Click *OK*.

13. Choose *Simulation - Run* to run the simulation. Alternatively, you can click the green arrow button located on the right of the ADE Explorer window to run the simulation.

© 1994-2020 All Rights Reserved.

14. After the simulation run is complete, click the blue up arrow button located to the left of *Setup,* under the Setup pane, The ADE Assembler window is displayed with *Setup* displayed under *Tests* in the *Data View* assistant, as shown below.

15. Click the right mouse button on *Setup* and select the *Create Test Copy* option from the context menu.

© 1994-2020 All Rights Reserved.

A copy of Setup by the name *Setup:1* is created, as show below.

16. Click twice on *Setup:1* and change the name to *AC_DC*.

Click the + signs to the left of *AC_DC* and *Analyses* to see that dc analysis is added under AC_DC as it was set under Setup.

17. Click on *Click to add analysis* text under dc analysis. The *Choosing Analyses* form is displayed.

18. In the *Choosing Analyses* form, select *ac* from the *Analysis* section.

19. Select *Start-Stop* for *Sweep Range* and type *1M* in the *Start* field and *10G* in the *Stop* field.

20. Select *Logarithmic* from the *Sweep Type* drop-down list.

© 1994-2020 All Rights Reserved.

21. Type *20* in the *Points Per Decade* field.

22. Click OK.

© 1994-2020
All Rights Reserved.

Note that ac analysis is added in the test, as shown below.

© 1994-2020
All Rights Reserved.

You now have two tests, Setup and AC_DC in ADE Assembler. You can create multiple tests in ADE assembler in the same manner. You can click on the down blue down arrow symbol to the left of any test to open the test in ADE Explorer, as shown below.

When you click an Up arrow, you switch from the single-test ADE Explorer environment to the multi-test ADE Assembler environment. When you click a down arrow, you switch from the ADE Assembler multi-test environment back to the ADE Explorer single-test environment. ADE Assembler can be thought of as the place where multiple ADE states are created and managed. In this environment, you can run all or any subset of the tests that are defined. When you switch to the ADE Explorer environment, you are in the single-test equivalent of ADE L.

© 1994-2020 All Rights Reserved.

© 1994-2020 All Rights Reserved.

3

Frequency Domain Analyses: Harmonic Balance

Overview of Simulation Capabilities

SpectreRF offers two different modes; Spectre and APS. Spectre uses traditional simulation techniques. APS makes the CMOS models more efficient and allows multi-threading for the device current evaluation. In addition, APS adds full parallel solving of all the matrices, thus speeding up the simulation even more. There is no loss in accuracy because there is no change in the actual equations that are simulated or in the options that control the accuracy of the simulation. These choices are available and are implemented for harmonic balance and periodic steady state simulations. Both large-signal harmonic balance and small-signal hbac, hbsp, and hbnoise analyses have the Spectre and APS implementations.

Large-Signal Harmonic Balance Overview

Harmonic balance is a large-signal analysis that solves in the frequency domain. It calculates the harmonics and mixing products of one or more inputs to the circuit. It takes into account all the large-signal effects in the circuit. Applications include measuring the harmonic content of a single input, calculating the large-signal IP3 of a power amplifier (with 2 input frequencies), and measuring the large-signal IP3 and IP2 of a transmit mixer (with 2 IF tones and an LO input). Harmonic balance can also provide a large-signal solution to the small-signal hbac, hbsp, and hbnoise analyses. More information on small-signal analyses is provided later in the chapter.

Harmonic balance is chosen for circuits that have near-sinusoidal signals and do not have high speed transitions. Although harmonic balance may work with circuits that have rapid transitions by setting lots of harmonics, it is likely that shooting (available in pss) will run faster. For LC oscillators and especially crystal oscillators, harmonic balance should be used. For ring oscillators, it is likely that shooting pss will be faster and more accurate. However, harmonic balance can be used on very nonlinear circuits including circuits with multiple frequency dividers.

© 1994-2020 All Rights Reserved.

The implementations of harmonic balance in the ADE Explorer/Assembler *Choosing Analyses* form differ somewhat. The *Choosing Analyses* form has three choices that have harmonic balance (hb) as a choice: hb, pss with the engine set to harmonic balance (hb), and qpss with the engine set to harmonic balance.

In hb and qpss when harmonic balance (hb) is selected as the engine, the harmonics of only the inputs and the mixing products are calculated. In pss, when harmonic balance is selected as the engine, all the harmonics of the inputs are calculated, even if the power in those harmonics is zero. For example, if there is an input at 2GHz and another at 2.1GHz, the harmonics of 100MHz are calculated. Note that in order to calculate through the fifth harmonic of the input at 2.1GHz, 105 harmonics need to be calculated. For this reason, pss is usually used for a single input frequency, and qpss or hb is usually used for multiple input frequencies.

The hb analysis does not require any distinction. It always calculates the harmonics of the inputs, and if multiple inputs are present, it also calculates the mixing frequencies. Also, the hb analysis has an auto mode where all you do is provide the input frequency. Everything else is automatic. Oscillator tuning mode is available where you specify a target frequency and a parameter to be varied, and then hb will tune the oscillator to that frequency. Usually, this is followed by a noise analysis, and this capability allows Monte Carlo simulations where the oscillator is tuned to the frequency, and the noise is measured at that frequency. In addition, hb has a mode where the gain compression is automatically determined. Using this capability speeds up compression analysis in Monte Carlo analyses. Multiple frequency dividers are supported with settable divide ratios for each signal path. This chapter focuses on the hb selection in the ADE Explorer *Choosing Analyses* form.

Harmonic Balance provides a solution in the frequency domain. Things that have direct frequency domain representations like capacitors, transmission lines, or S-parameter descriptions go directly into the frequency domain solution. Devices and other nonlinearities need to be evaluated in the time domain, with the ifft, and fft used to translate between the domains. This is an iterative process. Each successive iteration produces a more accurate solution. In other words, an exact solution is never attained. The iterations continue until the answer is “close enough”. Close enough is set by choosing an accuracy default in the *Choosing Analyses* form. Liberal, moderate, and conservative accuracy levels are provided. Choose the appropriate accuracy level based on your requirement. If you are calculating very small amplitude harmonics, you might need to choose conservative.

© 1994-2020 All Rights Reserved.

Example

Consider a behavioral low noise amplifier shown below:

© 1994-2020
All Rights Reserved.

The LNA is a behavioral LNA from rfLib. The properties list is shown below. Note that the gain is 20 dB, and the amplifier is only slightly nonlinear.

© 1994-2020
All Rights Reserved.

The input port (the source on the left side in the above schematic) has the properties shown below.

© 1994-2020
All Rights Reserved.

In general, give the signals names in the *Frequency name* property. In the example above, it is RF for the first frequency, and RF2 for the second. Giving names is required if you use the list of signals in the circuit in the harmonic balance *Choosing Analyses* form instead of specifying the frequencies. The frequency is set to a variable name called *frf* and the amplitude in dBm is set to the variable name *prf*. Setting variables in these properties makes it easy to change the frequency or amplitude without needing to change the schematic. The variable *frf* is set to 2.4G and the variable *prf* is set to -30 under *Design Variables* in the *Setup* pane of ADE Explorer, as shown below.

ADE Explorer is used to set up the different analyses by clicking on the *Analysis - Choose* menu, clicking on *Click to add analysis* text located under *Analyses* in the *Setup* pane, or by clicking the ChooseAnalysis icon ( ) located at the upper-right corner of the ADE Explorer window.

© 1994-2020 All Rights Reserved.

When you select *Analysis-Choose*, the *Choosing Analyses* form appears. At the top of the form, select the *hb* (harmonic balance) radio button.

© 1994-2020
All Rights Reserved.

In the *Choosing Analyses* form, the frequencies, the number of harmonics, oversample factor and accuracy are set. Oversample factor will be discussed later in the chapter. The *Fundamental Frequency* field can either have a number like 2.4G, or it can have the variable name that is used to set the frequency in the input port as shown above.

Run transient has three settings: *Decide automatically*, *Yes*, and *No*. *Decide automatically* means that the transient will be run before the harmonic balance algorithm. In this mode, the transient starts with a small number of periods of the fundamental frequency for Tone1 as the transient stop time. As the transient analysis runs, the waveform is analyzed to see if steady-state has been reached. When steady-state is reached, a Fourier transform is performed on the last cycle of the transient waveform, and this is used as the starting point of the harmonic balance iterations. If steady-state has not been reached in this small number of periods, the transient is extended in time up to a maximum of 250 periods for a driven circuit, and 500 periods for an oscillator. If steady-state has not been detected in the last period of the transient, other continuation methods will be tried in an attempt to achieve convergence.

Decide automatically also sets the number of harmonics in the *Choosing Analyses* form for *Tone1* to *auto* by default. The transient analysis waveforms at all the nodes in the circuit are analyzed, and based on the waveforms, the number of harmonics that are needed for the simulation to be accurate is set by the simulator. This can be manually overridden by setting a number of harmonics manually.

Setting *Run transient* to *Yes* requires you to enter a stop time for the initial transient analysis in the *Stop Time (tstab)* field. This mode allows the steady-state detection to be selected or not. If it is selected, whenever steady-state is detected in the transient analysis, it will terminate, run the fft, and start the harmonic balance iterations. In this mode, the stop time for the transient analysis cannot be automatically extended. If *Detect Steady State* is deselected, the transient will run to the specified stop time without checking for steady-state.

When *Run transient* is set to *Yes*, and the *Detect Steady State* is deselected, the number of harmonics field remains blank by default. You can either enter the text *auto*, or you can manually set the number of harmonics by specifying a number. When *auto* is set, the transient analysis waveforms at all the nodes in the circuit are analyzed, and based on the waveforms, the number of harmonics that are needed for the simulation to be accurate is set by the simulator.

Setting *Run transient* to *No* causes the harmonic balance algorithm to start from the DC solution without transient assist.

Dynamic parameters are supported in tstab just like in the transient analysis. Typically, only reltol is changed over the tstab interval with loose (large) values of reltol at the beginning of tstab and then tighter values toward the end to improve convergence. A single option is

© 1994-2020 All Rights Reserved.

supported in the GUI. Note that the default values are already fairly loose in the tstab interval by default.

When you have completed the *Choosing Analyses* form, click *OK* and run the simulation. You can run the simulation by selecting *Simulation - Netlist and Run*, or by selecting the green arrow on the right side of the ADE Explorer window.

When the simulation completes, select *Results - Direct Plot - Main Form*. The *Direct Plot Form* is displayed.

© 1994-2020
All Rights Reserved.

At any time, the next thing that needs to be accomplished is shown at the bottom of the form.

The *Analysis* section at the top of the form displays the list of different analyses that were run. In this example, only *hb* results are available.

The *Function* selection is where the type of data that is desired to be plotted is selected. *Power* is shown above.

To get the power based on the voltage in a net, *Net(specify R)* is chosen from the drop-down list.

Next select *Magnitude*, which gives power in Watts, dB10, which gives dB with respect to 1 Watt, or dBm, which gives dB with respect to 1 milliwatt. from the *Modifier* section. *dBm* is shown above.

The next step is to select the net in the schematic. The waveform window appears with the selected result in it. Both the input and output nets are shown. A marker has been positioned at the first harmonic in both traces. Note that the input level is almost exactly -30 as set in ADE Explorer, and the output is 20dB larger than the input as set in the behavioral LNA. The harmonics at about -350dBm represent the numerical noise floor of the simulation.

© 1994-2020
All Rights Reserved.

Harmonic Balance Solves For Cosines

In order to be compatible with other harmonic balance simulators, the convention in Spectre harmonic balance is to solve for cosines rather than sines. From a practical point of view, this means that the phase that is solved in harmonic balance is 90 degrees different compared to shooting. It also means that in order to get a zero degree solution in harmonic balance, the phase of the input source needs to be 90 degrees.

The circuit below is used for this example.

© 1994-2020
All Rights Reserved.

In the properties list for the input port, the phase has been set to 90 degrees. This is the phase of the sinusoid that is produced by the input source.

© 1994-2020 All Rights Reserved.

© 1994-2020
All Rights Reserved.

The signal *in1*, which is the input to the behavioral amplifier has been plotted below. The power in dBm is unaffected by the phase, but you can see that the phase that is calculated by Spectre is zero degrees. In harmonic balance, if you want a reference phase of zero, you must set the phase on the input source to 90 degrees.

Setting Harmonics Automatically

Harmonic balance can now set harmonics automatically for the signal that causes the most distortion. This is recommended in the general case. To enable this, select *Decide automatically* or *Yes* for the *Run Transient* selection in the Transient-Aided Options. This single action will cause a transient analysis to be run until steady-state is detected, and then from the transient analysis, the number of harmonics for Tone1 (when *Frequencies* is selected) or for the tone that has tstab enabled (when *Names* is selected).

Setting Harmonics Manually

Consider an amplifier with a single input. In this case the input and its harmonics need to be calculated. In order to get the correct solution, both positive and negative frequencies need

© 1994-2020 All Rights Reserved.

to be calculated. With harmonic balance, input is required as to how many harmonics should be calculated. If you set 3 harmonics, then -3. -2, -1, 0, 1, 2, and 3 times the input frequency needs to be calculated, as shown below.

The number of harmonics to be calculated is (2*number of harmonics) +1. The runtime and memory consumption are proportional to the number of unknowns (all the harmonics at all the nodes) that need to be solved. In this case, seven unknowns need to be solved at each node.

In order to make sure that you have enough harmonics, start with an estimate based on the power level and the harmonic content of the input to the circuit. If the input is sinusoidal at fairly low power, try about 3 harmonics. If the amplitude is large, try about 7 harmonics. Next, run the simulation and plot the results. Position a marker at the desired measurement. Now increase the number of harmonics, and re-run the simulation and re-plot the waveform. If the measurement did not change, then the original estimate was enough, and you might be able to reduce the number of harmonics. Use the smallest number of harmonics that are required in order to have a minimum runtime.

If the input is a pulse wave at high amplitude, start at about 15 harmonics and an oversample factor of 4. Oversample factor will be discussed later in the chapter. The simulator takes the fft of the input signal, and uses multiple sources in series with the values and phases set to the value calculated by the fft.

When square waves are present in the circuit, the minimum number of harmonics should be set to the period of the square wave divided by the risetime of the square wave. Very sharp edges require many harmonics in order to be accurate in the time domain. If the time-domain waveform is less important than the frequency domain content, then a smaller number of harmonics can be specified along with an oversample factor of 4 or 8. For more information, refer to the Oversample Factor on page 68.

When piecewise linear waveforms are used as power supplies to ramp the power up at time zero, specify only two points. The first point should have time set to zero, and the voltage should be about 80% to 90% of the actual supply voltage. The second point should have the time set equal to about half the period of the operating frequency, and the voltage set to the supply voltage. If the simulation time exceeds the second time in the PWL setup, it will retain that value for the rest of the simulation. More importantly, for any SpectreRF large-signal

© 1994-2020 All Rights Reserved.

analysis, the system does not become periodic until after the last timepoint in the PWL file. If you put a point at 1 second and the power supply voltage, the system will need to be simulated for one full second in the tstab interval using the transient algorithm before the harmonic balance simulation can run. This can take a long time.

When periodic piecewise linear inputs are used, the more non-sinusoidal the waveform is, the more harmonics you need and also the higher oversample factor needs to be in order to accurately simulate the system. Oversample factor is explained in the following section. Start with an estimate of how many harmonics would be required to represent the waveform in the frequency domain, and then re-run the simulation with more harmonics to see if things changed. Oversample factor also needs to be increased. Sine waves need oversample=1. Square waves need 4 or 8. The more nonsinusoidal the waveform is, the higher the oversample factor needs to be. Note that even with a sinusoid applied, the currents may be very nonsinusoidal. Think of the currents in a diode mixer with a sinusoidal LO input. In this case, an oversample factor of 4 or 8 is likely to be required.

Up to two non-sinusoidal sources are allowed in the circuit. The first can be any periodic signal type, like pulse, exponential, and periodic piecewise linear. The second non-sinusoidal source is limited to being a pulse waveform. The rest of the inputs must be sinusoidal.

Oversample Factor

Note that fft and ifft are used to translate between time and frequency domains for the nonlinear evaluation. It is the nature of the ifft and fft to require more samples than the absolute minimum required in the time domain to get a correct answer for the harmonics in the frequency domain when the waveform is nonsinusoidal.

© 1994-2020 All Rights Reserved.

The circuit below has a square wave source on the top, and a sine wave source on the bottom. The load for both is a resistor.

There is no nonlinearity at all in the circuit. Therefore, the harmonics of the sinusoid and the square wave can be analyzed using the transient analysis and the DFT function in the calculator. The DFT function uses an fft algorithm to calculate the peak value of each

© 1994-2020 All Rights Reserved.

harmonic. The transient waveforms are shown below. Both signals have exactly 1 volt peak amplitudes.

In the following figure, you can see that the fft of the sine wave is essentially the same for 4, 16, 64, and 256 samples of the transient waveform. The waveforms are shown in the top-left

© 1994-2020
All Rights Reserved.

sub window. The number of samples can be read in the legend at the top-left corner of each sub window.

All the readings are in pico-dB or femto-dB. Sine waves do not need extra timepoints to calculate the correct amplitude for the harmonic.

© 1994-2020
All Rights Reserved.

Shown below are the ffts of the square wave with 4, 8, 16, 32, 64, 128, and 256 samples. The number of samples can be read at the upper-left corner of each sub window in the legend.

Note that with four samples, the value is 0dBV, which is 1 volt. With 256 samples, the result is 2.09777dB. This is almost 2.1dB larger than with four samples and is obtained just by increasing the number of samples. With 32 samples, you can get almost the right answer. This is a factor of eight times the number of samples required to calculate two harmonics. In other words, if two harmonics are set, an oversample factor of at least eight is required to get the correct answer for the square wave.

© 1994-2020 All Rights Reserved.

When 8192 points are used, only a minor change is observed as shown below.

© 1994-2020
All Rights Reserved.

Trading off Harmonics and Oversample Factor

The risetime of the above square wave is 1/500th the period of the square wave. Technically, the period divided by the risetime or 500 harmonics are required to obtain the correct answer. The hb solution for the waveform (from an ifft) and in the frequency domain are shown below.

The large number of harmonics requires a long runtime when real circuits are simulated. Because of the large number of harmonics, shooting pss is likely to run faster than hb for this level of accuracy.

© 1994-2020 All Rights Reserved.

If accuracy in the time domain can be compromised a bit, setting a relatively small number of harmonics with an oversample factor of 8 can get reasonably close to the solution above. In the below example, 5 harmonics with an oversample factor of 8 have been used.

Note that the answer in the frequency domain is only a few thousandths of a dB different than the answer with 500 harmonics. The waveform in the time domain (from an ifft) is very different than the transient waveform. If you only need an answer in the frequency domain, setting a small number of harmonics along with a high oversample factor will result in much faster runtimes with little compromise in accuracy in the frequency domain when non-sinusoidal waveforms are present.

The basic strategy with non-sinusoidal waveforms is to start with an estimate of how many harmonics might be required, and set the oversample factor to 8. Run the simulation. Reduce the number of harmonics by about 50% and run the simulation again. If the answer does not change, reduce the number of harmonics again. Do the same with oversample. You are looking for the lowest runtime that also produces the correct answer, which will usually occur with a relatively large number for oversample factor, and a relatively small number of harmonics.

Every circuit is a little bit different in terms of the number of harmonics and oversample factor, but the principles are the same.

© 1994-2020 All Rights Reserved.

Two Input Frequencies

Consider an amplifier with two input frequencies. Because the circuit is nonlinear, intermodulation distortion is created. A lot more harmonics are produced by the circuit and need to be solved in the simulation. As a result, the runtime is longer and more memory is required. The harmonics to be calculated can be seen graphically, as shown below. Three harmonics are specified for both signals. The horizontal axis represents the first frequency and has symbols at -3*input frequency, -2*input frequency, -1*input frequency, the DC level, +1*input frequency, +2*input frequency, and +3*input frequency. Similarly, the vertical axis has symbols at the frequencies of the harmonics for the second input signal. The first symbol, 45 degrees from the origin, is the mixing product `Frequency1` + `Frequency2`. The figure below shows several different mixing products.

Note that the actual frequencies are not represented well by the chart. If F1 is 1GHz, and F2 is 1.1GHz, then F1+F2 is 2.1GHz and this is the first point at +45 degrees. F1-F2 is 100MHz, and is represented by the first point at -45 degrees.

This is called a rectangular or box cut, which is the default setting in harmonic balance. Note that ((2*number of harmonics on tone 1) +1) * ((2*number of harmonics on tone 2) +1)

© 1994-2020 All Rights Reserved.

harmonics need to be calculated. In this case 49 harmonics need to be calculated at each node of the circuit.

Note that the diagonal corners are sixth order terms. For example, the upper-right corner is 3*F1 + 3*F2. Since it is a sixth order term, it is likely that the amount of power that the circuit generates is small, and therefore can be removed from the solution space, thereby speeding up the simulation with minimal loss in accuracy.

Frequency Cuts

Note that widefunnel, crossbox, and crossbox_hier cuts are intended for multi-divider mode. Do not use these frequency cuts for a normal simulation. These cuts will be covered in the Multi-Divider Mode section below.

© 1994-2020 All Rights Reserved.

To access frequency cuts, set harmonics to Select and then choose the appropriate frequency cut. This is shown below for the diamond cut. The parameters for the different cuts and the harmonics that are retained are explained below.

© 1994-2020 All Rights Reserved.

SpectreRF has a diamond cut available where the high order terms can be removed from the simulation. The solution space is shown below for a 4th order diamond cut. The solutions in red are in the solution space. The solutions in black are not. The figure below assumes that 7 harmonics are specified for `Frequency 1`, and 3 for `Frequency 2`.

It is called a diamond cut because graphically, it looks like a diamond. The rectangular cut would have 15*7 or 105 harmonics in it. This cut has 39 harmonics in it. Instead of calculating 105 harmonics at each node, 39 harmonics are calculated at each node, therefore, the simulation runs considerably faster and takes less memory. All the fourth order and lower order terms are in the solution. The fifth order and higher order terms are excluded. As long as there is not much power in those harmonics, the solution will remain accurate and will run much faster than the default rectangular cut.

The diamond cut is useful for circuits where the distortion is similar in the circuit from all the input frequencies. An example is an IP3 simulation for a power amplifier. Because the power level of both inputs is the same, both signals cause similar distortion.

The order of the cut can be set in the ADE Explorer *Choosing Analyses* form. Setting the order of the cut is similar to setting the number of harmonics. The more the distortion in the circuit, the larger the order needs to be for accuracy. Start with an estimate of what the circuit might produce, and run the simulation. Take the measurement. Raise the order, re-run, and re-plot. If the measurement changed significantly, then the order needs to be raised again. If the measurement did not change, try a smaller order. Use the smallest order that produces accurate results.

© 1994-2020 All Rights Reserved.

Funnel cut is also available. Funnel cut is a diamond cut with the addition of the harmonics on the axis for all the tones. This cut is useful for systems where there is a difference in the distortion from the different inputs. One example would be a mixer with a large amplitude LO, and an RF tone. In this case, because the LO power is large, all the harmonics of the LO are desired, but the mixing products are limited to lower order. This reduces aliasing with the addition of only a small number of harmonics. A diagram is shown below.

© 1994-2020
All Rights Reserved.

Three Input Frequencies

Now imagine a mixer with an LO and two RF signals. Here, there are multiple planes of harmonics that need to be calculated, as shown in the figure below. Only 2 harmonics of the RF tones and one harmonic of the LO are shown in the figure so the planes can be visualized.

Note that ((2*number of harmonics on tone 1) +1) * ((2*number of harmonics on tone 2) +1) * ((2*number of harmonics on tone 3) +1) harmonics need to be calculated. If three harmonics were specified for all 3 tones, 7*7*7 or 343 harmonics need to be calculated at each node. Four input frequencies is a practical maximum for most circuits.

© 1994-2020 All Rights Reserved.

Diamond Cut With Three Frequencies

When the diamond frequency cut is applied, the red solution shown below is calculated when the maximum order is set to two. On the zero plane for the LO, a diamond that goes to the second harmonic of both tones is included. In the -1 and +1 planes for the LO, the diamond is drawn with the order reduced by one. When the number of LO harmonics is larger, which is the general case, on the 0 (zero) LO plane, the diamond goes through the order specified in the `maximorder` parameter. For the +1 and -1 planes, the order is reduced by one. For the +2 and -2 planes, it is reduced by one more. This continues on the rest of the planes.

© 1994-2020
All Rights Reserved.

Now imagine that there were more LO harmonics. More planes would be created. If the maximum order was set to 4, on the center plane, the included elements would go through the fourth order. For the first plane above and below, harmonics through the third order would be calculated. For the second plane above and below, harmonics through the second order would be included. For the third plane above and below, only the first harmonic would be calculated. For any other planes, no harmonics would be calculated.

You can see from the figure that fewer harmonics are calculated when the cut is applied and that the diamond is extended vertically. Fewer harmonics are calculated in the upper and lower planes than in the center plane. In this way, many harmonics are pruned from the solution space, and the simulation is sped up in the process. Two harmonics for the LO is artificially small in order to understand the concept. Usually, between 3 and 15 harmonics are chosen in most circuits.

Funnel Cut

Note that in the following example, because of the high LO power, many LO harmonics would be created in the circuit, and therefore, might cause incorrect results if the higher harmonics of the LO were not included in the solution. Funnel cut takes the diamond cut, and adds harmonics along the axes for all the inputs up to the maximum specified by the number of harmonics for each tone. This is shown in the figure below. Note that the signal names on the

© 1994-2020 All Rights Reserved.

axes are changed from the previous figure. This is done to allow the diagram to fit on the page.

Note that the harmonics along the axes only are added. In this case, the LO harmonics are added. The funnel cut is provided for circuits like mixers where the distortion is different from the different inputs. The idea is that the maximum order of the diamond might be 4 or 5 which allows accurate calculation of the mixing products and still calculating enough harmonics of the signal(s) that causes relatively more distortion. If the diamond cut were used, a higher maximum order is necessary to allow the high order harmonics of the high distortion signal to be calculated which includes many more harmonics to be solved than the funnel cut.

© 1994-2020 All Rights Reserved.

Axis Cut

A fourth cut is available called Axis cut that only calculates the harmonics of the inputs themselves. No mixing products are calculated at all. This cut is only useful for getting a quick idea of the amplitudes of the individual tones. It is shown in the figure below.

Multiple Frequency Dividers

This mode is intended for expert users who have an estimate of layout parasitics including parasitics from the substrate available.

Today's devices contain multiple radios. The large amplitude LO signals can cause leakage problems from the LO in one radio to the front end of another radio. Multiple dividers are now supported in harmonic balance to allow simulations of leakage from one radio into another.

© 1994-2020 All Rights Reserved.

Multiple divider mode is intended for the case where multiple frequency dividers are being driven from separate sources at different frequencies. If you have multiple frequency dividers being driven from a single source, do not use multiple divider mode.

The dividers must be driven using sources. Multi-divider mode is not supported for oscillator-driven frequency dividers.

To achieve acceptable accuracy, divider simulations using HB require a large number of harmonics on each input frequency. In the default mode, multi-divider simulations would require excessive computational time and require a large amount of memory to run. Therefore special frequency cuts that are specific to the case of multiple signal paths where the signal paths have weak coupling between them but each path requires a large number of harmonics are provided.

The funnel cut may be accurate enough, however three new cuts are available for multi-divider mode. These cuts require less memory and faster execution compared to the other frequency cuts. The author recommends starting with crossbox_hier before using the other two cuts.

© 1994-2020 All Rights Reserved.

Crossbox_hier

Crossbox_hier is supported for two tones only.

The number of harmonics on Tone 1 sets the width of the spectral components on the X Axis. The number of harmonics on Tone 2 sets the height of the spectral components on the Y Axis. Axisbw sets the width of the spectral components outside the central box on both the X and Y Axes. Maximorder sets the width of the central box on both the X and Y Axes.

As the axisbw setting is raised, more mixing harmonics are calculated which improves accuracy at the cost of simulation time and memory. This should be determined in the same way as setting the number of harmonics in HB. Start with one, and run. Plot the output and measure the amplitude of the desired harmonics. Raise axisbw and run again. If the measured harmonics changed, then raise axisbw again. Use the smallest axisbw that produces stable results.

© 1994-2020 All Rights Reserved.

Maximorder sets the width of the central box. The process is similar to setting axisbw above. Start with an estimate of the highest order mixing products that exist in your design, and calibrate by varying maximorder. As with axisbw, larger values produce more accuracy at the cost of simulation time and memory.

Crossbox

The frequency cuts are the same as crossbox_hier above. The parameters behave the same way as above.

The crossbox frequency cut has aliasing that is unrelated to the number of harmonics set on the individual input tones. This aliasing is controlled by the property called minialiasorder. The default value for this parameter is 600 which is likely to be very conservative. Setting smaller values will raise the numerical noise floor while reducing the runtime. Use the smallest value that produces accurate results.

© 1994-2020 All Rights Reserved.

Widefunnel

Widefunnel is similar to crossbox except the shape of the cut is different for the harmonics near the origin.

As in crossbox, minialiasorder needs to be set in the same way. The behavior of axisbw and maximorder is the same as in crossbox, but the shape of the internal area is diamond shaped instead of a box. The process for setting these is the same as in crossbox_hier.

Aliasing

The reason for setting the number of harmonics and the order of different cuts is because of aliasing. If the actual system produces power outside the solution in the higher harmonics, that power will appear in the solution you specified. In the example above with the axis cut,

© 1994-2020 All Rights Reserved.

because no mixing products are calculated at all, all that power shows up as increased power somewhere in the harmonics that are actually calculated.

Aliasing needs to be checked, as can be seen in the following figures.

The figure above displays the spectral calculation of an amplifier with 900MHz and 920MHz applied at the same level. Note the level of the harmonics at 880MHz, 900MHz, 920MHz, and 940MHz. In this solution, two harmonics were set for both tones.

When the number of harmonics in the solution is raised to three for each input, the aliasing is less because there is less power in the uncalculated harmonics of the system. Note that in this case, the measurements for the harmonics at 900MHz and 920MHz changed only by 0.01dB. If you only need a measurement of the output power at the main output frequencies, 2 harmonics are enough. The measurements at 880MHz and 940MHz have changed by

© 1994-2020 All Rights Reserved.

0.268dB and 0.263dB respectively. This might or might not be accurate enough for an IP3 measurement depending on your accuracy requirement. The solution is shown below.

In order to check if the solution has enough harmonics, even more harmonics need to be run. Below is the solution with five harmonics on each tone. Note that the levels have changed by 0.01 dB for both tones at 880MHz and 940MHz. Three harmonics are enough in this case.

© 1994-2020 All Rights Reserved.

Although the solution changed slightly with more harmonics, it is difficult to justify the extra runtime for a 0.01 dB change.

When the harmonics are raised to nine, the solution changes by less than 0.001dB as compared to five. To repeat, three harmonics is enough in this case. There was no significant change when the number of harmonics was raised above three.

© 1994-2020 All Rights Reserved.

The solution with nine harmonics is shown below.

Below is a slightly different view of harmonics. Harmonic balance is supported in pss, where all of the harmonics starting from zero through the highest harmonic are calculated.

Note: This is not suggested for most applications. This is just to understand the concepts only.

The first view shows the solution with slightly more harmonics than the number needed to calculate the upper third order product at 940MHz. The circuit is an amplifier with inputs at 900MHz and 920MHz.

© 1994-2020 All Rights Reserved.

The third order products are at 880MHz and 940MHz.

Note that there are significant harmonics from about 380MHz through 520MHz. These harmonics are not produced by the circuit. These harmonics are produced because the power in higher harmonics is significant, and that power shows up in the solution with the number of harmonics set up for this run. The take away from this is that if you do not calculate enough harmonics in the solution space, random and unpredictable errors are introduced in the solution specified by you.

When the number of harmonics are raised, the solution changes. The solution below has 205 harmonics and appears to have no aliasing. Note that the first order terms at 900MHz and

© 1994-2020 All Rights Reserved.

920MHz have only changed slightly, but that the third order terms have changed by a significant amount.

However, when the number of harmonics is raised to 210 (only 5 more harmonics), the figure changes. Here, it is apparent that aliasing is still going on, and it is a bit difficult to infer which harmonics are real and which are produced by aliasing.

© 1994-2020 All Rights Reserved.

This is shown below.

© 1994-2020
All Rights Reserved.

It is only when 1005 harmonics are taken that the aliasing stops because the levels of the higher harmonics are on the order of the numerical noise floor.

Gain Compression Analysis

Gain compression analysis is available in the harmonic balance *Choosing Analyses* form. Compression measurements can be made with either a port or a voltage source as the input to the circuit. Compression analysis does not require manually setting any input sweeps, and has dedicated direct plot functions to plot voltage or power at the compression point and the compression curves.

Gain compression is a measure of gain reduction resulting from circuit nonlinearity. Spectre allows you to measure either transducer power gain or voltage gain compression. Multiple levels of compression can be specified, if desired. In Xdb compression analysis, Spectre

© 1994-2020 All Rights Reserved.

sweeps the input power (or voltage) automatically to arrive at the desired compression level. Compression is defined as shown below.

The gain reference for compression measurement can be either the gain at small input amplitude as shown above, or the gain reference can be the maximum at the peak above. The diagram above shows the gain, but the actual power (or voltage) in the sweep can also be plotted. Small-signal analyses can also be specified and will be run at the compression point only.

Compression can measure power or voltage compression. For a power compression measurement, the input must be a port, and the output power is measured through a resistor, port or current probe. For a voltage compression measurement, the input is a vsource and The output is taken across a pair of nodes.

Multiple levels of compression can be specified in a run. In the *compression Levels (dB)* field, type a list of compression values separated by spaces.

© 1994-2020 All Rights Reserved.

Rapid Mode is also provided. This mode supports only a single compression level. In this mode, you provide an estimate of the input power or voltage at the compression point. Then, three runs are made. One at low power, one at the level you specified, and a third that is slightly lower then the compression point. Results are then extrapolated based on these three simulations. This mode works well for Monte Carlo and Corners where a large number of simulations are required.

Gain compression analysis stores the following information:

1. The compression point.
2. Gain and power compression curves.
3. The HB simulation results at the compression point.

Direct Plot functions are provided for the results.

Small-signal analyses can be set up and run, and will be run only at the gain compression point.

Large-Signal S-Parameters

Large-signal S-Parameters can be set up in the harmonic balance Choosing Analyses Form and Direct Plot functions are available after the simulation runs. This can be done for amplifiers or for circuits that translate frequencies. Set the number of harmonics manually to be enough for the largest power level of the sweep of input power.

To use LSSP, click the LSSP check box in the hb Choosing Analyses Form. Click the Select button to the right of the Ports field and select the input and output ports in the circuit. Click the Choose button to the right of the Load Harmonic field, and select the frequency of the output. This allows LSSP measurements for systems that convert frequencies. Now define an input sweep using the Sweep check box above the LSSP check box.

On the first power level of the sweep, S11 and S21 are measured. Then the input port is disabled and the output port is set to the power measured in the forward S21 measurement. This allows S22 and S12 to be measured. This process is repeated until the large-signal S-Parameters are measured for all the sweep levels.

When this completes, LSSP measurements can be done using the Direct Plot Form. The S-Parameters, impedance of the circuit, and VSWR can be plotted at the sweep values.

© 1994-2020 All Rights Reserved.

Sine Representation of Square Wave

In harmonic balance, a square wave is approximated by a series of sinusoids at the odd harmonics. The actual waveform that the circuit sees is shown from a transient simulation in the waveform window shown below. The top trace has the first and third harmonic. The second trace contains all the harmonics through the 11th harmonic. The third trace contains all the harmonics through the 19th harmonic. The bottom trace has 25 harmonics. Note that as more harmonics are added, a closer approximation is made to the square wave. The flat part of the waveform gets flatter and the edge gets steeper.

Note that this is done for all non-sinusoidal waveforms by taking the fft of the waveform specified through the harmonic number and then stacking the voltage sources in series with the values set to the value from the fft.

© 1994-2020
All Rights Reserved.

Convergence

Like all harmonic balance simulators, an iterative approach is used to solve the circuit equations. In an iterative approach, there is always another iteration that causes the solution to become more accurate. You stop iterating as soon as the error in the solution becomes small. As a result, the solution is not exact.

To indicate progress in the Spectre output, two metrics are provided.

- `Resd_norm` measures the absolute error in the solution.
- `Delta_norm` measures the change from the last iteration.

Convergence is achieved when the sum of the currents at all the nodes is near zero, and there is only a small motion between the last iteration and the current iteration. When both `resd_norm` and `delta_norm` are less than one, the simulation has converged and the iterations stop.

The absolute accuracy is affected by the product of `reltol * residualtol`. The change that is allowed form iteration-to-iteration is affected by the product of `reltol *lteratio * steadyratio`.

Reltol is a global option, which is an option that affects all analyses that are run. This option can be set in ADE Explorer by selecting *Simulation - Options - Analog*. The default for reltol depends on the setting of `liberal`, `moderate`, or `conservative`, whether the circuit is driven or is an oscillator, and the number of inputs to the circuit. The *Residualtol* option is located just below the *reltol* option in the form, and is a multiplier to reltol. Normally residualtol is not used in SpectreRF. Lteratio depends on the setting of `liberal`, `moderate`, or `conservative` for the errpreset option.

Errpreset

Errpreset is shown in the ADE implementation section. Normally, the option *lteratio* is not set away from the default for harmonic balance. *Steadyratio* is a harmonic balance option, and defaults to 1.0. Setting a larger *steadyratio* value allows a larger change from iteration-to-iteration and still allows convergence. Following are the default values for *reltol*, *lteratio*, and *steadyratio*.

1 input driven circuit	Reltol	Lteratio	Steadyratio
Liberal	1e-3	3.5	1.0
Moderate	1e-3	3.5	1.0
Conservative	1e-4	10	1.0

© 1994-2020 All Rights Reserved.

multiple input driven circuit or oscillator	Reltol	Lteratio	Steadyratio
Liberal	1e-3	3.5	1.0
Moderate	1e-4	3.5	1.0
Conservative	1e-5	10	1.0

Note that the values for the *reltol* option are the maximum values for the frequency domain iterations only. For transient assist, the value of the *reltol* option in the global options form is used. If the *reltol* option is set to a value smaller than the values in the table, that value will be used. If the *reltol* option is set to a value higher than the values in the table, the values in the table will be used.

In some cases, the `resd_norm` (a measure of absolute error) in the Spectre output window will be below one, but the delta norm (a measure of the change between iterations) remains above one. In this case, the solution varies from iteration to iteration more than the amount allowed. This is usually caused by the small-amplitude harmonics changing relatively a lot, but since they are small, allowing a large delta is acceptable. Look at the delta norm to see the range it is in, and then set the *steadyratio* option slightly larger than that and re-run the simulation.

Iteration Limit

All iterative approaches have the possibility of not converging. In hb, there is a default limit of 100 iterations. This can be changed by setting the `maxperiods` option.

Note: If the iteration limit is reached without convergence, the solution point for the last iteration is written to the data file and is available for plotting. Make sure that the simulation actually converged by checking the Spectre output window. If you are running a sweep, the sweep will stop when the error is incurred.

© 1994-2020 All Rights Reserved.

The message is located at the end of the output, and is shown below.

```
/usr2/rdavis/demos/ic61/spectrerf_aps_workshop/simulation/DVCO_RFX_tb2/spectre/schematic/ps
File  Help                                                                    cadence

********** iter = 98 **********
Damping Factor is 0.3
Delta Norm=1.61e+03  at node I24.R1:1  harm=(1 0)
Resd Norm=2.34e+03  at node I12.C4.node3  harm=(1 0)
Frequency= 5.1047e+09 Hz, delta f= 5.94e+07

********** iter = 99 **********
Frequency Damped
Damping Factor is 0.1
Delta Norm=5.21e+03  at node I24.R1:1  harm=(1 0)
Resd Norm=2.88e+03  at node I12.C4.node3  harm=(1 0)
Frequency= 5.0026e+09 Hz, delta f= -1.02e+08

********** iter = 100 **********
Delta Norm=3.49e+03  at node I24.R1:1  harm=(1 0)
Resd Norm=2.31e+03  at node I12.C2.node3  harm=(1 0)
Frequency= 5.1144e+09 Hz, delta f= 1.12e+08

Error found by spectre during Harmonic Balance Steady State Analysis `hb'.
    ERROR (SPCRTRF-15278): Maximum number of iterations reached. Simulation failed to converge.



*************************************************
Fundamental frequency is 5.11438 GHz.
*************************************************

CPU time=13 s

Analysis `hb' was terminated prematurely due to an error.

Notice from spectre.
    49 notices suppressed.
    49 warnings suppressed.

modelParameter: writing model parameter values to rawfile.
element: writing instance parameter values to rawfile.
outputParameter: writing output parameter values to rawfile.
designParamVals: writing netlist parameters to rawfile.
primitives: writing primitives to rawfile.
subckts: writing subcircuits to rawfile.

12
```

You can see from *resd_norm* at the last iteration, that the error is significant. (it is much greater than one.) This answer cannot be trusted. To allow more iterations, set the `maxperiods` option to a number greater than 100.

© 1994-2020 All Rights Reserved.

Number of Non-Sinusoidal Sources

Up to two non-sinusoidal sources are allowed in the circuit. The signal that causes the largest number of harmonics or is periodic piecewise linear in the circuit should be *Tone1* when *Tones* are set to *Frequencies*. *Tones* is a choice in the *Choosing Analyses* form. If *Tones* is set to *Names*, enable tstab in the list for this signal. The second non-sinusoidal source is limited to being a pulse waveform

All the rest of the sources in the circuit must be sinusoidal. When *Frequencies* is selected, there is a limit of four input frequencies in the circuit. *Names* does not have this limitation, but four input frequencies is a practical upper limit because the number of harmonics to solve explodes.

When *Names* is selected, the sources in the circuit need to have entries in the *Frequency name 1* or *Frequency name 2* properties in the signal sources in the circuit. When assigning names, make sure that if you have multiple inputs at a single frequency, like there might be for a differential circuit, the names should be the same. This enables SpectreRF to consider the two signals as one. If you have frequencies that are integer multiples of each other, these should have the same name too. In this case, the actual frequency will be the lower of the two frequencies. Make sure that enough harmonics of the low frequency are specified to incorporate enough harmonics for the high frequency input to prevent aliasing.

Harmonic Balance Starting Point

Harmonic balance has traditionally started from the DC solution for the first iteration in the frequency domain. This is not the default for SpectreRF. The default is to run a short transient until steady-state is reached, and then switch to harmonic balance where the number of harmonics is selected automatically based on the transient analysis waveforms.

Starting from a DC point is a reasonable starting point for driven circuits. The input signal causes harmonics to be created and solved in the frequency domain. Transient aided harmonic balance is also available. This is accomplished by selecting Decide automatically or Yes for run transient. If Yes is selected, setting a simulation time in the stop time property is needed. Transient aided runs a transient analysis on the signal that you select for the time specified, and then runs a single period of the input. A Fourier transform is performed, and the result is used as the starting point for the frequency domain iterations. Because the starting point is closer to the actual solution, it usually requires fewer frequency domain iterations at the expense of running the transient analysis at the start. It is frequently faster to run a short transient and then iterate in the frequency domain. The only way to determine which is faster is to try both ways and see which one finishes first. Only one signal can have transient assist. When *Tones* is set to *Names*, the signal with tstab set to *yes* can have transient assist. Choose the signal that produces the largest amount of distortion for transient

© 1994-2020 All Rights Reserved.

assist. When *Tones* is set to *Frequencies*, only *Tone1* can have transient assist. *Tones* is an option in the *Choosing Analyses* form.

Transient assist can also be used to improve convergence. It works because the starting point in the frequency domain is closer to the real solution than the DC solution, which is the default. Anytime an iterative approach is used, the closer the starting point is to the real solution, the higher the probability of convergence.

In some cases, very lengthy transient simulations are needed to produce convergence. There are two ways you might use to shorten this time. The first way is to specify a filename in the *writehb* option on the first run, and then use the *readhb* option on subsequent analyses. The *writehb* option writes out every relevant parameter to a file so that when *readhb* is specified, the simulation can start from the file. The second way to reduce the time is to save the state near the end of the transient assist (tstab) interval, and then use that state to restart the transient assist with a much shorter simulation time before going into the frequency domain iterations. This is accomplished using the *savefile*, *saveperiod*, and *savetime* options.

Both of these ways allow circuit values to be changed when restarted, but the topology of the circuit cannot be altered. When the topology is altered, the transient assist needs to start over. If a value is changed, the change causes an instantaneous discontinuity in the solution and because of this, if the change is too large, there will be convergence failures.

More Capabilities

Save and Recover

Save and recover writes and restores the full internal state of the transient analysis that is used in the tstab interval so that a circuit can be restarted from that state to save time in a simulation that has a long startup time, or in the case where a longer tstab is needed for the transient assist. For now, savefiles that are made using the transient analysis are not usable as a recover file in harmonic balance. The savefile must be created from the tstab interval in harmonic balance or pss.

To save a file, set the *savefile* option to a filename. If you are working from a netlist, the file will be created in the directory you are running from. If you are in the ADE environment, the file is located in the `netlist` directory. This is located in the *simulation/<circuit_name>/<simulator_name>/<view_name>/netlist* directory. The simulation directory defaults to your home directory, but it can also be set by the administrators of the Cadence tools. In most cases, this directory will be shown in the project directory of the *Setup* - *Save*Options menu in ADE Explorer. If you are unable to locate this directory, contact your system administrator to find the location. Spectre is the simulator name, so if you had a circuit called `abc`, and you

© 1994-2020 All Rights Reserved.

started ADE Explorer from the schematic view, the `netlist` directory would be located at *~/simulation/abc/spectre/schematic/netlist*.

Next, set either *saveperiod* or type in a list of times in the *savetimes* field. Do not specify both at the same time. If tstab is set to 50n, *saveperiod* is set to 10n, and the *savefile* option is set to *save_file*, when the simulation reaches 10n in the tstab interval the file *save_file* is created in the `netlist` directory. When 20n is reached, *save_file* is overwritten with the data at 20n. This continues until the last 10n interval is reached in the tstab interval.

Alternatively, you can type 10n 25n 40n 50n in the *savetimes* field. This would create four files beginning with *save_file* and ending with the time for that specific point. The exact format of this file changes occasionally, so look in the `netlist` directory to get the actual filename you want to start from.

To recover, you specify the filename you want to recover from, and set tstab to a larger time than the time in the file. You can also change the list of times in the *savetimes* option to save more times after the time you recovered from.

Writehb and Readhb

Setting *writehb* causes the entire harmonic balance solution to be written to a file that can be used later. When readhb is set to a previously saved *writehb* file, the solution is read from the file, and then follow-on small-signal analyses can be performed. Note that *readhb* and *writehb* cannot be set at the same time.

Readhb can also be used as a starting point for when more input frequencies are supplied to the circuit. If you have a 2-tone solution, and then you add a third tone, The two tones in the solution are used as a starting point for the 3-tone solution which must iterate to achieve convergence. Because the starting point contains the solution for two signals, this is a better starting point than using transient aided where the solution to only one tone is available for the first iteration.

Sweeps and Restart

Sweeps are available both in the hb *Choosing Analyses* form and from the parametric tool in ADE Explorer. Sweeps automatically use the previous solution as the starting point for the next member of the sweep. If convergence difficulties are encountered, the sweep parameter will be automatically stepped from the value that converged to the value that needs to be measured. When Monte Carlo encounters convergence difficulties, several things are tried to achieve convergence. If you want to disable these automatic features, set the restart option to *yes*.

© 1994-2020 All Rights Reserved.

Itres

Itres controls the precision of the solution at the first iteration in the frequency domain. Harmonic balance starts with the DC solution as the starting point which causes the solution at the first iteration to be inaccurate. Because of this, the first solution is calculated without much precision (number of digits in the solution) in order to save time. Follow-on iterations have more precision. The final solution has full precision. Note that if you use transient-aided harmonic balance, the starting point in the frequency domain can be quite good, and so itres should be manually set to 1e-2. This forces enough digits in the mantissa to be solved so that the solution differs from a solution with all the digits of the mantissa by less than 1 part in 100. If the circuit is nearly linear (for example an LNA) increasing the precision by setting itres to 1e-2 can also reduce the total number of iterations. Regardless of the setting of itres, as the iterations progress, the precision is automatically increased so that the final solution is accurate. The default is 0.1, which allows a 10% error on the first iteration.

Freqdivide

If the circuit has a frequency divider, specify the divide ratio in the *freqdivide* property. This needs to be an integer. Make sure that oversample factor is set in this case as discussed earlier. Also, make sure that transient-assisted is used, and set the transient part to at least the time needed for the divider to count through its sequence once. The effect of this is that the fundamental frequency is divided down by the freqdivide ratio. Make sure that enough harmonics are specified so that at least the fifth harmonic of the input signal is calculated. If *freqdivide* is set to four, make sure that at least 20 harmonics are specified for that signal.

Oscillators

For oscillators, a DC starting point is not a good starting point. The frequency and amplitude of the harmonics needs to be solved and the DC state has no information at all about this. There is no signal source, so the oscillator usually just does not start. The default in ADE Explorer is to estimate the oscillatory frequency and amplitude, and then run a transient analysis until steady-state is reached. Upon reaching steady-state, the number of harmonics needed for the oscillator are calculated by evaluating the waveform at every node in the oscillator, and then running harmonic balance to solve for the frequency and amplitude at each node. This is a good starting point for LC, crystal, or mems oscillators.

If the oscillator is a ring oscillator, apply the initial conditions to force one stage of the ring to a high or low state. Set *Run transient* to *Decide automatically*. If desired, turn off *Calculate initial conditions automatically*, which will only work for a conventional feedback oscillator.

© 1994-2020 All Rights Reserved.

When using transient assist manually (called the tstab interval) for oscillators, set *Run Transient* to *Yes*. Specify a time for tstab, on the order of twenty periods of oscillation. Select the check box for *Detect Steady State*. This will enable the transient, and automatically switch to harmonic balance when steady-state is reached. In this mode, tstab cannot be lengthened automatically like it can be when *Decide Automatically* is selected.

In many cases, it is desirable to see the startup waveform, which is not saved by default. If you want to see the startup behavior, select *yes* for the *Save Initial Transient Results (saveinit)* option.

Dynamic parameters are available during the tstab interval. Dynamic parameters allow changing option settings during tstab and are typically used to reduce the accuracy (which speeds up the simulation) at the beginning and then tighten the accuracy at the end to improve convergence.

When oscillators are simulated, a node or pair of nodes (if it is differential) needs to be declared to the simulator. This pair of nodes is used to get an estimate of the oscillation frequency at the end of the transient assist period. At the end of the transient assist period, 5 periods of the estimated operating frequency are simulated. In these 5 periods, the simulator looks for the frequency divided output at every node in the circuit and if it finds this behavior, it automatically adjusts the frequency. The frequency is estimated based on these last 5 cycles. A Fourier transform is performed on the last single cycle, and this is the starting point in the frequency domain. If frequency dividers are in the circuit, the estimated frequency of the oscillator is recommended to be specified in the *Choosing Analyses* form, and the frequency divide ratio should be specified in the freqdivide option.

Solving the frequency in addition to the amplitudes and phases adds another unknown to the solution set, so it usually takes more iterations to reach a solution compared to a driven circuit.

Pinnode

In a driven circuit, all the phases and amplitudes of the harmonics are defined by the input signal. There is no input for an oscillator, so a pinnode must be selected. The first harmonic phase of that node is pinned (held constant), and then the amplitude of all the nodes and relative phases of all the other nodes are calculated using the pinnode phase being held constant. SpectreRF has an algorithm that automatically selects the pinnode, and it is usually reliable. If convergence is an issue, try setting the pinnode manually in the options form to one of the nodes in the resonator.

To set the pinnode manually, select the *Options* button at the bottom of the hb *Choosing Analyses* form and select the convergence tab (*IC61 only*). In the options field, do not enter a slash(/) at the beginning of the expression. Just enter the hierarchical node name similar to

© 1994-2020 All Rights Reserved.

this example: `I12.I1.osc_resonator`. In the schematic, IC61 shows the instance name of the block in the tab for that block, so it is a bit easier to get the hierarchical names.

Probe-Based Method

The normal method of solving for an oscillator allows the amplitudes, phases, and frequencies to be solved simultaneously. A second method called the probe-based method iterates for the frequency solution in the outer loop and the amplitude and phase solution in the inner loop. It places a voltage source between a pair of nodes you specify, which should be in the resonator. If you have a single-ended oscillator, specify one node only. If the second node is left blank, it will be connected to the global ground node automatically. The voltage source converts the oscillator into a driven circuit, which converges easier. The source amplitude, phase, and the frequency are adjusted until the current in the source is near zero. Do not use the probe-based method and the pinnode option together, since the probe-based method also requires a pinnode to be selected. *Harmonic Index* and *Magnitude* do not need to be specified for the probe-based method. The *harmonic index* defaults to 1, which is suggested, and the *Magnitude* defaults to a small amplitude. If you specify *Magnitude*, make sure that the magnitude you specify (in volts RMS) is smaller than the amplitude of the actual oscillations, or the method itself will fail.

Oscillator Tuning Mode

Tuning mode adjusts a parameter in the circuit to produce the set frequency target. This is done automatically when tuning mode is enabled without setting any sweep parameters or interpolation of the resulting curves. When the tuning frequency is reached, any small-signal analyses like noise are run. This allows the simulator to tune the oscillator to a specified frequency and then make a noise measurement. This is useful in Monte Carlo analysis to see how the oscillator performs with process variations. The parameter to be tuned can be a variable, temperature, or a specified device parameter. Oscillator tuning mode is also supported in the pss analysis using the shooting or harmonic balance engines. See Chapter 4, “Single Input Large and Small-signal Analyses,” for details.

In this mode, the target frequency to tune to is the frequency that is specified in the harmonic balance form as the fundamental frequency. You specify a parameter that is to be varied to achieve the desired frequency. This can be a variable, a device parameter, or temperature. When the analysis runs, the oscillator will be tuned to the desired frequency, and then all the small-signal analyses will be run.

Direct plot functions have been added to plot the tuning parameter.

© 1994-2020 All Rights Reserved.

Semi-Autonomous

Semi-autonomous is provided to allow the simulation of circuits that have periodic sources and oscillators in the same circuit. It is only supported in the hb *Choosing Analyses* form. Typical applications are for when you have a receiver with an RF tone and an oscillator for the LO so that all the oscillator noise is taken into account for the receiver noise simulation. Another application is for identifying the phase noise and spurious response of an oscillator when power supply ripple is applied.

The simulator combines the driven and oscillator capability in one analysis. To set it up is similar to a multi-tone simulation, just select the *Oscillator* check box and specify the estimate of the oscillator frequency in the *osc!* section of the form. The probe-based method for oscillators is not available for semi-autonomous simulations.

Things to try to Achieve Convergence (Driven Circuits)

Try running transient-assisted by setting the tstab option. Ensure that the signal that causes the most distortion is in *Tone 1* (when *Frequencies* is selected) or has tstab enabled (when *Names* is selected). Transient assist can help on any circuit.

If there is an S-Parameter block in your circuit, make sure that causality is set to *fmax* on every instance of the nport. Starting with MMSIM 11.1 this is the default. The effect of this is to enforce causality for the DC and tstab interval. If this selection is not made, even the DC solution can be affected, sometimes producing results that can be larger than the power supply voltage. Setting causality to *fmax* makes sure that the DC and tstab intervals have a causal model. Refer to the Simulating S-Parameters chapter for a description about the nport component. Also, make sure that every port of the nport has a connection to something in the circuit. Leaving a terminal open can be very bad for convergence.

Look at the startup waveform to see if it is settled. Make sure that tstab is long enough. Look at the starting waveform, and ensure that the amplitude is reasonable for your circuit. To improve convergence, the waveform must be pretty close to settled at the end of the tstab interval where the Fourier transform is made.

Use enough harmonics. If your system produces square waves, make sure that at least five harmonics are calculated. In many cases, many more harmonics than that are required.

Try setting the oversample factor to a higher value. This should be set to four or eight for circuits that have square waves. Setting it to two or four improves the accuracy of the Fourier transforms used in harmonic balance (hb) even for nearly sinusoidal circuits.

Try setting itres to 1e-2. This forces a solution that is more precise on the first iteration, thus improving the chance of convergence.

© 1994-2020 All Rights Reserved.

Try more maxperiods if `resd_norm` is small (near one). This indicates that the circuit is almost converged, and might converge more if more iterations are allowed. The default maxperiods is 100.

If the residual norm is below one, but the delta norm is greater than one, set *steadyratio* just a little higher than the value in which the delta norm is running. The effect is to allow more delta from iteration to iteration and still allow convergence. Only do this if the `resd_norm` is less than one, which indicates that the absolute accuracy check is good.

Try moderate accuracy. Sometimes, loosening the tolerances allows a circuit to converge. Note that this does reduce the accuracy of the solution the simulator solves for.

Things to try to Achieve Convergence (Oscillators)

Try everything listed above. In addition:

Try setting *reltol* to 1e-5 and *vabstol* to 3e-8. These options can be found in the *Simulator Options* form in ADE Explorer. Select *Simulation - Options - Analog* to open the *Simulator Options* form. Making *reltol* and *vabstol* smaller makes the initial transient much more accurate by reducing the error caused by iterating to each solution and the numerical integration error. By providing a more accurate estimate of the frequency and amplitude of the waveform, convergence becomes easier.

Try setting a pinnode manually. While the algorithm that is used in SpectreRF is usually reliable, if the oscillator does not converge, set the pinnode to one of the nodes in the resonator.

Select the *Calculate initial conditions automatically* check box if you have an LC oscillator. This causes an initial estimate of the frequency and amplitude to occur at the start of the tstab interval. Add transient assist so that the harmonics can be calculated for the starting point in the frequency domain.

Try the probe based method. This method adds a voltage source to the circuit so it becomes a driven circuit. Because the circuit is driven, convergence is easier. The frequency amplitude and the phase of the source are adjusted so that the current of the source becomes zero. When this happens, the source has no effect on the circuit.

Try setting an initial condition (current) in the inductor of the resonator. Choose a value that approximates the steady-state peak value for the oscillatory condition.

© 1994-2020 All Rights Reserved.

Implementation in ADE Explorer

Setting Frequencies, Harmonics, and Oversample

Setting Harmonics and Transient Assist Automatically

The default in ADE Explorer is to run transient assist until steady-state is just reached, and then choose the number of harmonics for the first tone automatically, and then proceed with solving in the frequency domain.

© 1994-2020 All Rights Reserved.

© 1994-2020
All Rights Reserved.

The defaults are highlighted above. This configuration is recommended for all hb analyses.

```
/servers/scratch03/kamal/spectrerf131/testcase131/docs
File  Edit  View  Help                                              cadence

Steady-state is detected after 1.25 ns.

=====================================
`hb': time = (1.25 ns -> 1.66667 ns)
=====================================

    hb: time = 1.26 ns      (2.5 %), step = 2.083 ps     (500 m%)
    hb: time = 1.281 ns     (7.5 %), step = 2.083 ps     (500 m%)
    hb: time = 1.302 ns    (12.5 %), step = 2.083 ps     (500 m%)
    hb: time = 1.323 ns    (17.5 %), step = 2.083 ps     (500 m%)
    hb: time = 1.344 ns    (22.5 %), step = 2.083 ps     (500 m%)
    hb: time = 1.365 ns    (27.5 %), step = 2.083 ps     (500 m%)
    hb: time = 1.385 ns    (32.5 %), step = 2.083 ps     (500 m%)
    hb: time = 1.406 ns    (37.5 %), step = 2.083 ps     (500 m%)
    hb: time = 1.427 ns    (42.5 %), step = 2.083 ps     (500 m%)
    hb: time = 1.448 ns    (47.5 %), step = 2.083 ps     (500 m%)
    hb: time = 1.469 ns    (52.5 %), step = 2.083 ps     (500 m%)
    hb: time = 1.49 ns     (57.5 %), step = 2.083 ps     (500 m%)
    hb: time = 1.51 ns     (62.5 %), step = 2.083 ps     (500 m%)
    hb: time = 1.531 ns    (67.5 %), step = 2.083 ps     (500 m%)
    hb: time = 1.552 ns    (72.5 %), step = 2.083 ps     (500 m%)
    hb: time = 1.573 ns    (77.5 %), step = 2.083 ps     (500 m%)
    hb: time = 1.594 ns    (82.5 %), step = 2.083 ps     (500 m%)
    hb: time = 1.615 ns    (87.5 %), step = 2.083 ps     (500 m%)
    hb: time = 1.635 ns    (92.5 %), step = 2.083 ps     (500 m%)
    hb: time = 1.656 ns    (97.5 %), step = 2.083 ps     (500 m%)

Notice from spectre during Harmonic Balance Steady State Analysis `hb'.
    Auto harmonic calculation has chosen 5 harmonics for tone-1.


==============================
     Harmonic balance
  hbhomotopy=tone (1-tone)
==============================
Important HB parameters:
    RelTol=1.00e-04
107  >                                                   L1      C1
```

Spectre will calculate the number of harmonics that are needed and report the number in the spectre output file, as highlighted above.

© 1994-2020 All Rights Reserved.

Tones = Frequencies

© 1994-2020
All Rights Reserved.

1. For sinusoidal sources, set *Oversample Factor* to 1.
2. For square waves, set *Oversample Factor* to 2, 4, or 8. Use the smallest number that provides accurate results.
3. If there are multiple sources with a common integer multiple frequency in your circuit, specify the highest common frequency. For example, if 1GHz and 1.5GHz are present in the circuit, set *Tone1* or *Tone2* frequency to 500MHz. In this case, make sure you take enough harmonics of 500MHz to get an accurate result at 1.5GHz.

Important

Specify the signal that causes the most distortion in the circuit as Tone1. This is the tone that can have transient assist, and selecting the signal with the most distortion as tone 1 improves the accuracy of the simulation.

Tones = Names

1. Select a line in the sources list.

   The fields (highlighted below) are automatically populated with the values of the selected line in the sources list.

2. Change the number of harmonics under the *Mxham* column (auto in the figure above).

© 1994-2020 All Rights Reserved.

3. Change the oversample factor under the *Ovsap* column (1 in the figure above).
4. Set only one of the inputs to have transient assist by setting *Tsta*b to *yes*.

   Note: Make sure you enable tstab for the signal that has the largest amount of distortion in the circuit. This will make harmonic balance more accurate.

5. If there are multiple sources in your circuit that have integer frequency relationships, set the *Frequency name 1* property in all the sources in the schematic to the same name. This will cause all the sources to be treated as solving for harmonics of the highest common frequency. For example, if 2.4GHz and 3.6GHz were present in the circuit, a single-tone simulation would be run that solves the harmonics of 1.2GHz. Make sure that you use enough harmonics so that the enough harmonics of 3.6 GHz are solved to prevent aliasing errors.

Dynamic Parameters

To set the dynamic parameters:

© 1994-2020 All Rights Reserved.

1. Open the hb *Choosing Analyses* form.

© 1994-2020
All Rights Reserved.

2. Choose *Yes* for *Run Transient?*.
3. Specify the stop time for tstab in the *Stop Time (tstab)* field.
4. Select *yes* for *Save Initial Transient Results (saveinit)*.
5. Specify a value in the *Fundamental Frequency* field.
6. Select *Oscillator*.
7. Click *Select* to the right of the *Oscillator node+* field.
8. Select a net in the schematic that has an oscillator signal.
9. Select *conservative*.
10. Select *Dynamic Parameter*.
11. Choose *Parameter vector* from the drop-down list.
12. Specify the name of the option that you want to vary in the tstab interval in the *Parameter name* field. The example shows *reltol*.
13. Specify *0* (zero) in the *Time* field.
14. Specify a reasonable value for the option. This example steps reltol from 1e-2 to 1e-7 every 20 nsec in the tstab interval. Note that 1e-2 is a very large value for reltol.
15. Click *Add*.
16. Add other time-value pairs as appropriate.
17. Click *OK*.

© 1994-2020 All Rights Reserved.

18. Run the simulation. The setting for the dynamic parameter appears in the Spectre output window.

© 1994-2020 All Rights Reserved.

Multiple Frequency Dividers

1. To simulate circuits with multiple frequency dividers, set *Tones* to *Frequencies*.

© 1994-2020 All Rights Reserved.

2. Select *Yes* from the *Run Transient?* drop-down list, and specify a tstab of at least two complete divider cycles for the longest divider output period.
3. Select *Multi-divider Mode*.
4. Specify the input frequency to the dividers in the *Fundamental Frequency* fields.
5. Set the harmonics manually for both tones. Specify enough harmonics so that a minimum of five harmonics are solved at the input frequency of the divider.
6. Set *Oversample Factor* to two or four. See Oversample Factor on page 68.
7. Specify the divider ratio for each tone in the *Freq Divide Ratio* fields. In order to run frequency dividers, tstab will be run on each tone in the list separately.
8. Choose *Select* from the *Harmonics* drop-down list.
9. Choose *funnel* cut.
10. Specify *5* in the *MaximOrder (int)* field.

Diamond Cut

This is used when the number of significant harmonics produced by the circuit are similar for all the inputs. This is usually the case for a large-signal IP3 measurement on an amplifier.

In the *Choosing Analyses* form:

1. Choose *Select* from the Harmonics drop-down list.

© 1994-2020 All Rights Reserved.

2. Select *diamond*.
3. Type a reasonable value in the *MaximOrder (int)* field. The value is usually between 3 and 7.

Funnel Cut

This is used when the number of significant harmonics produced by the circuit are different for one or more inputs. This is usually the case for a large-signal IP3 measurement on a mixer (The LO usually produces more harmonics than the RF input signals).

In the *Choosing Analyses* form:

1. Choose *Select* from the Harmonics drop-down list.
2. Select *funnel*.
3. Type a reasonable value in the *MaximOrder (int)* field. The value is usually between 3 and 9.

© 1994-2020 All Rights Reserved.

Sweeps

You can sweep up to three things in the hb *Choosing Analyses* form.

- Each can be a variable, temperature, component parameter, or model parameter.
- Choose what you want to sweep from the drop-down list.
- Set the start and stop range for the sweep in the *Start* and *Stop* fields.
- You can add points off the grid in the *Add Specific Points* field.

© 1994-2020
All Rights Reserved.

Freqdivide

If you set freqdivide, make sure that you have at least five harmonics at the input frequency as an absolute minimum. You must also set the *Oversample Factor* between 2 and 8.

Normally, when there are square waves in the circuit, for accurate results for the transient waveform, the number of harmonics needs to be at least the period of the square wave divided by the rise or fall-time, whichever is shorter.

© 1994-2020 All Rights Reserved.

Compression Analysis

Compression analysis automatically finds the compression point without needing to define a sweep of the input power. Multiple compression levels can be set in the *Compression Level (dB)* field. Separate the levels with a space.

First, set a reasonable number of harmonics that is enough for the simulation to be accurate at the compression point.

To enable compression analysis:

1. Select the check box to the right of *Compression* in the *Choosing Analyses* form.
2. If desired, select *Rapid Mode*. In this mode you must specify the approximate input referred compression level. This mode is most useful for Monte Carlo and Corners runs where there are many simulations.

© 1994-2020 All Rights Reserved.

3. Select *Power* or *Voltage* from the *Gain Definition* drop-down list.

© 1994-2020
All Rights Reserved.

4. Select either the *Small-Signal Gain* or the *Maximum Gain* as the *Gain Reference*.

© 1994-2020
All Rights Reserved.

5. Set the compression level. This can be a list of values separated by a space for normal mode, and must be a single value for Rapid Mode. Also set the harmonic number of the

© 1994-2020 All Rights Reserved.

output. To view the harmonic frequencies, click *Choose* to the right of the *Output/Load Harmonic* field.

© 1994-2020
All Rights Reserved.

6. Select the output source and output load.
7. If required, enter a value in the *Backoff* field. When you specify a value (in dB) in this field, there can only be a single value of compression. The simulator runs the compression analysis and then lowers the power from the compression point by the specified value. Next, it runs HB and all hb<xx> analyses specified at that reduced input power level.
8. Run the simulation.
9. Open the *Direct Plot Form*. To plot the compression results, select *xdb*.

Large-Signal S-Parameters

In the harmonic balance setup, make sure you set the number of harmonics manually to a number large enough to capture an accurate solution at the highest power level of the sweep. Auto harmonics should be avoided because it will set harmonics based on the lowest power level in the sweep.

1. In the hb Choosing Analyses form, select LSSP check box.
2. Click the *Select* button to the right of the Ports field and select the input and output ports in the circuit.
3. Click *Show* to the right of the Load Harmonic field and select the output frequency from the list.

© 1994-2020 All Rights Reserved.

4. Click the *Sweep* check box. In this example, the input power is set by the variable prf, which is swept from -20 to 0 dBm with 1 dB steps. Note that when LSSP is enabled, small-signal analyses like hbnoise cannot be run.

© 1994-2020
All Rights Reserved.

5. Run the simulation.
6. When the simulation completes, select *Result - Direct Plot - Main Form*.
7. In the Direct Plot Form, select what you want to plot and the type of chart to plot on. This example shows S11 plotted on a Z-Smith chart.

© 1994-2020
All Rights Reserved.

S11 results are plotted in the waveform tool, as shown below.

In the waveform tool, select *Marker - Snap Tracking Cursor*. The cursor will snap to the data points. The sweep power level is shown below the measurement.

Oscillator Additions

In the *Choosing Analyses* form, select *Names* or *Frequencies*.

© 1994-2020
All Rights Reserved.

Tones = Frequencies

1. If you have an oscillator, select *Oscillator*.

   Tone 1 will automatically change to *osc!*.

   The oscillator node location in the circuit only needs to have a signal on it.

2. Select *Decide automatically* from the *Run transient?* drop-down list. This will cause an estimate of the oscillator frequency and amplitude to be run, then the transient runs in the tstab interval, but only to the point of steady-state. Once steady-state is reached, a Fourier transform is calculated, and the frequency domain iterations begin.

3. Set the frequencies, number of harmonics, and oversample factor.

   The *Fundamental Frequency* field should have a value between 0.5 and 1.5 times the actual frequency of oscillation. If *Run transient?* is set to *Decide automatically* or *Yes*, *auto* is set for the number of harmonics. At the end of the transient in the tstab interval, the number of harmonics will be set automatically based on the waveforms in the tstab interval.

© 1994-2020
All Rights Reserved.

4. Select *conservative* for oscillators.

Note: You should always select *conservative* for oscillators.

Tones = Names

1. If you have an oscillator, select *Oscillator*. The system will display a popup window that reminds you to provide an estimate for the oscillation field.

The oscillator node location in the circuit only needs to have a signal on it.

2. Specify a value in the *Expr* field that is between 0.5 and 1.5 times the actual frequency of oscillation.

© 1994-2020
All Rights Reserved.

3. Select *Decide Automatically* from the *Run transient?* drop-down list. This will cause tstab to be run only until steady-state is reached, where a Fourier transform is calculated, and the frequency domain iterations in harmonic balance begin.

4. *Decide automatically* also evaluates the Fourier transform at the end of the tstab interval, and sets the harmonics based on the actual spectrum. The word *auto* appears in the *Mxharm* field.

5. Select *conservative* accuracy for oscillators.

Note: You should always select *conservative* for oscillators.

Oscillator Tuning Mode

Oscillator tuning mode is provided to tune the oscillator to a desired frequency and then run the selected small-signal analyses. This is useful for sweeps and for Monte Carlo simulations. To run an oscillator tuning analysis, do the following:

© 1994-2020 All Rights Reserved.

1. Open an oscillator circuit, and start ADE Explorer. Open the hb *Choosing Analyses* form, and select hb analysis.
2. Fill out the form as usual, except for the following:
   a. In the *Fundamental Frequency* field, type the desired frequency target.
   b. Click the check box next to *Enable tuning mode analysis*.

© 1994-2020 All Rights Reserved.

3. Select whether you want to tune a variable, temperature, or a component parameter. The example below shows a variable.

© 1994-2020 All Rights Reserved.

4. Enable the desired small-signal analyses.
5. Run the analyses. When the simulation completes, open the *Direct Plot Form* from ADE Explorer.

Probe-Based Method

Note: Earlier, this used to be called the two-tier method.

In the *Choosing Analyses* form:

1. Choose *Frequencies* or *Names*.

© 1994-2020
All Rights Reserved.

2. Select *Use the probe-based solution method.*

© 1994-2020
All Rights Reserved.

Note: When the probe based method is used, the only choice is to set the *Run transient?* option to *No*.

3. Select an oscillator node that is inside the feedback system for *Pinnode+*.

4. If you have a differential circuit, select the differential node at the mirror location in the feedback system for *Pinnode-*.
5. Leave the *Harmonic Index* and *Magnitude* fields blank.

Semi-Autonomous

Semi-Autonomous allows an oscillator and a periodic source to be simulated at the same time.

In the *Choosing Analyses* form, select *Names* or *Frequencies*.

© 1994-2020
All Rights Reserved.

Frequencies

1. Select *Oscillator*.

2. Select the number of tones as desired.
3. Type an estimate for the *Fundamental Frequency* in the *osc!* column.
4. Type the frequencies for the periodic sources in the other columns.
5. Set the number of harmonics and the oversample factor as required. If *Run transient?* is set to *Decide automatically* or *Yes*, then *auto* is entered for the number of harmonics in the first tone only. This can be set manually, if desired. The number of harmonics for the rest of the tones must be set manually.
6. If there is a frequency divider after the oscillator in the circuit, set the divider ratio in the *Freqdivide Ratio for Tone 1* field. Remember to set oversample factor and enough harmonics when this is set.

© 1994-2020 All Rights Reserved.

7. If you want to reduce the number of harmonics set, choose *Select* from the *Harmonics* drop-down list and set *MaximOrder*.

8. Select *conservative*.

9. If you want to view the startup waveform, select *yes* for *Save Initial Transient Results (saveinit)*.

10. Specify the oscillator nodes in the circuit.

© 1994-2020
All Rights Reserved.

Names

1. Select *Oscillator*. The system will display a popup window that reminds you to supply an estimate for the *Expr* field.

© 1994-2020
All Rights Reserved.

2. Specify the frequency in the *Expr* field.

3. Select Decide automatically from the *Run Transient?* drop-down list. This will cause Spectre to run a transient in the tstab interval until steady-state is reached. Then, a Fourier transform is calculated, and the frequency domain iterations begin.

4. Set the harmonics and the oversample factor as required. If *Run transient?* is set to *Decide automatically* or *Yes*, then *auto* is entered for the number of harmonics in the first tone only. This can be set manually, if desired. The number of harmonics for the rest of the tones must be set manually.

5. Click *Change, or* select another field, or use the <*Tab*> key.

6. If there is a frequency divider after the oscillator in the circuit, set the divider ratio in the *Freqdivide Ratio for Tone 1* field. Remember to set oversample factor and enough harmonics when this is set.

© 1994-2020 All Rights Reserved.

7. If you want to reduce the number of harmonics set, choose *Select* from the *Harmonics* drop-down list.

8. Select *Conservative*.

9. If you want to view the startup waveform, select *yes* for *Save Initial Transient Results (saveinit)*.

10. Specify the oscillator nodes in the circuit. This node can be any node in the circuit where the oscillations are present.

Commonly Used Harmonic Balance Options (Driven Circuits and Oscillators)

The *Harmonic Balance Options* form deliberately omits uncommonly used options. In the unlikely case that you need options that are available in pss-harmonic balance, you will need to determine the `option name=value` pairs and enter this information in the *additionalParams* field in the *Misc* tab. The easiest way to get these `option name=value` pairs is to set the option in a dummy pss analysis, use ADE Explorer to create the netlist, and then highlight the text in the netlist reader. This text can be entered in the *additionalParams* field by clicking the center mouse button.

The options are divided into five tabs: *Convergence*, *Accuracy*, *Output*, *Reuse*, and *Misc* (miscellaneous).

The options are shown in the order that they appear in the *Harmonic Balance Options* form in ADE Explorer.

© 1994-2020 All Rights Reserved.

Convergence tab

Maxperiods

This specifies the maximum number of iterations for the hb algorithm. The default is 100. If this option is set less than 100, it will be ignored.

© 1994-2020
All Rights Reserved.

Itres

This option specifies the precision for the solution of the first iteration in harmonic balance. For subsequent iterations, the precision is increased so that when the iterations end, full precision is used. The default is 0.1, which allows up to 10% error in the first iteration. If you have a linear or near linear circuit like an LNA, lower itres to the 1e-2 to 1e-4 range so that the solution for the first iteration is more precise. Fewer iterations will be required for the solution to converge.

pinnode

This option is only available for oscillators. Type the pathname to the pinnode in this field. and example is I3.resonator. Do not use a slash (/) at the beginning of the name.

save osc ppv

This option is no longer used.

Ic

Initial conditions can be specified graphically by selecting *Simulation - Convergence Aids - Initial Condition* in ADE Explorer. Initial conditions can also be specified from a file using the *readic* property. Capacitors and inductors have the initial condition properties in the property list for the component. For capacitors, this is an initial voltage across the capacitor, and for inductors, it is an initial current in the inductor. The default is to observe all the initial conditions in the DC analysis that is used as the time-zero timepoint in the tstab interval. The initial conditions force a voltage or current to be present in the time-zero solution. The initial conditions are released for the rest of the transient simulation in the tstab interval. The *ic* option controls the initial conditions that should be observed in the time-zero timepoint. *all* is the default. *dev* means that only the initial conditions on capacitors and inductors are observed. *node* means that only the initial conditions on a node are observed. *dc* means that no initial conditions are observed. The example below shows the ic option set to *node*.

© 1994-2020 All Rights Reserved.

Skipdc

In some cases, the time-zero timepoint DC analysis does not converge. Instead of stopping the simulation, *skipdc* allows the tstab simulation to continue using an assumed solution for the time-zero timepoint. The default is *no* and a DC analysis is run to get the initial timepoint. *yes* means skip the DC solution, and proceed directly to the tstab simulation. All the nodes with initial conditions specified start at the initial condition value. Nodes with batteries start at the battery voltage. Nodes with no initial conditions start at zero volts. For `skipdc=yes`, the signal sources start as specified immediately in the tstab simulation. *sigrampup* uses the same assumptions for the starting voltages as *yes* does, but the start time is set to negative one tenth of the stop time for the tstab interval. At this time, the signal source time-varying part starts at zero and linearly ramps up to the full value at time = zero. After time = zero, the sources have the full amplitude time-varying part. The example below shows the *skipdc* option set to *sigrampup*.

Readic (Shooting and Harmonic Balance)

This specifies an ascii file that contains two columns to be read as initial conditions. The first column is the node name. The second column is the voltage value. If the entry in the *readic* field does not start with / (slash), the entry is located in the netlist directory. To find the netlist directory, select *Setup- Simulator/Directory/Host* in ADE Explorer. Look in the *Project Directory* field for the location of the simulation directory. Navigate to that directory and then to the `<Circuit Name>/spectre/<schematic or config>/netlist` directory. You can also click ( ) and browse to the directory. An example is shown below.

Readns

This specifies an ascii file with the same format as an ic file that is used as nodesets for the time-zero DC solution. Nodesets do not force a voltage to be held for the time-zero solution. Instead, they are a way of speeding up the time-zero calculation. As a suggestion, set the *write* option and the *readns* option to the same filename. The *write* option writes the time-zero solution to a file. When this is used as a starting point, many fewer iterations are needed for the time-zero point to converge. An example is shown below.

© 1994-2020 All Rights Reserved.

Cmin

cmin can be used to improve convergence in the tstab and shooting intervals for shooting and for the tstab interval for harmonic balance. If a value is set for *cmin*, a capacitor with this value is added to every node with the other terminal of the capacitor connected to the global ground node. If a 10f to 50f capacitor is added, this prevents instantaneous changes from occurring from timepoint to timepoint, thus improving the convergence at the cost of adding non-physical capacitors to the circuit. An example with 10 femtoFarads is shown below. Note that if 10 is entered, a 10 Farad capacitor is added from every node to ground. Remember the multiplier in the entry. An example is shown below.

Accuracy Tab

The *Accuracy* tab is split into two categories, *TIME STEP PARAMETERS* and *INTEGRATION METHOD PARAMETERS*.

© 1994-2020
All Rights Reserved.

Maxstep

This is the maximum allowable timestep for the transient analysis in the tstab interval.

Tstabmethod

This option controls the integration method in the tstab interval. The default value is determined by the setting of *errpreset* and whether the circuit is driven or is an oscillator. For *moderate* and *conservative* with a driven circuit, the default is *gear2only*. For *liberal* it is *traponly*. Generally, *gear2only* is preferred because of the absence of trapezoidal ringing inherent in the *trap* (trapezoidal) method. The default for *tstabmethod* is *traponly* for oscillators. *traponly* is used for oscillators because with the defaults, the tstab interval uses relatively loose convergence options in order to speed up the simulation. This causes relatively long timesteps which for the *gear2* method cause noticeable numerical damping. *traponly* does not numerically damp, therefore, it provides a better estimate of the oscillating frequency and amplitude. An example with *tstabmethod* set to *gear2only* is shown below.

© 1994-2020 All Rights Reserved.

Output Tab

Annotate

This option is currently not implemented.

Save

The default is *allpub*. This saves all the public voltages at all levels of the hierarchy in the schematic. It excludes the internal nodes of the device models. *all* adds the internal nodes of all the devices. *lvl* saves all the nodes including the internal nodes of the devices through the level of hierarchy set in the *nestlvl* option. *lvlpub* is like *lvl*, but it does not save the internal nodes. *selected* saves only the nodes that are specifically saved. In ADE Explorer, this is accomplished by selecting *Outputs - To Be Saved - Select On Design* and then selecting the nodes and terminals specifically in the schematic. At the netlist level, this is accomplished by using a save statement with a list of names to be saved.

Nestlvl (Shooting and Harmonic Balance)

If *save* is set to *lvl* or *lvlpub*, this controls the maximum level of hierarchy to be saved. If *nestlvl* is *1*, only the top level is saved. If *nestlvl* is 2, the top level and the next level down

© 1994-2020 All Rights Reserved.

are saved. The value for *nestlvl* can be any integer. The example below shows *lvlpub* selected that saves two levels of hierarchy.

Compression (Shooting and Harmonic Balance)

Normally, this should not be used for RF simulation. It is not digital compression. For RF simulation, where the input is sinusoidal, the size of the results file will normally double. It is useful only for circuits that are predominantly square wave.

Reuse Tab

© 1994-2020
All Rights Reserved.

Writehb

This option specifies that the full internal state of the harmonic balance analysis be written out to the file specified in the option. An example is shown below.

```
writehb    hb_state_file
```

Readhb

This option specifies the file to be read in as a starting point for the next harmonic balance analysis. If nothing is changed in the circuit, then only one iteration is performed and small-signal analyses can be run with a much faster time for the hb analysis. Changes can be made to the circuit or analysis options as long as the topology of the circuit stays the same. The changes introduce a discontinuity at the beginning of the hb analysis, which might be large enough to cause hb not to converge.

An example of *readhb* is shown below.

```
readhb    hb_state_file
```

Saveclock

saveclock saves the state of the simulation in the tstab interval at the clock time interval in seconds specified by *saveclock*. When the clock time has passed in the tstab interval, the file specified by the option *savefile* is created. When subsequent clock time intervals have passed, the file is overwritten. Use only one of the save options at a time. The example below shows *saveclock* set to three minutes.

```
saveclock    180
```

Saveperiod

saveperiod saves the state of the simulation in the tstab interval at the simulation time interval in seconds specified by *saveperiod*. When the simulation time has passed in the tstab interval, the file specified by the option savefile is created. When subsequent simulation

© 1994-2020 All Rights Reserved.

time intervals have passed, the file is overwritten. Use only one of the save options at a time. The example below shows *saveperiod* set to 25 nanoseconds.

Savetime

savetime is a list of times in the tstab interval where the state of the simulation is written out. The list should be specified with spaces between the entries. The information is written out with the filename specified in the *savefile* option with extensions for the time after that. Use only one of the save options at a time.

The example below shows four times in the tstab interval to save.

Savefile

This is the file name to write out the tstab state information to. If you do not specify a filename that starts with the slash (/) character, the file is stored in the *netlist* directory. To locate the *netlist* directory, first select *Setup* - *Simulator/Directory/Host* menu in ADE Explorer, and read the path to the project directory. This is the path to the *simulation* directory. In the simulation directory, navigate to the `<circuit_name>`*/spectre/<schematic or config>/ netlist* directory.

Note: Do not specify a relative path name like *./save_file/run1*.

Recover (Shooting and Harmonic Balance)

recover specifies the file name to recover the tstab simulation from. If *saveperiod* or *saveclock* are used to make the savefile, then the same name as specified in *savefile* is used. If *savetimes* has a list of times specified, then several files are created at the times

© 1994-2020 All Rights Reserved.

specified in the list. Click (...) and browse to the netlist directory where the savefiles are shown.

Select one from the list, and click *Open*. This adds the full path to the *recover* option, as shown below. Also note that in the example below, the *savefile* option is still set, and times after the restart times are specified. This is specifically allowed. As long as tstab is 200n or larger, the files at 175n and 200n will be added and can be reused later.

© 1994-2020
All Rights Reserved.

Now change tstab to a later time. In this case, 200n is set.

© 1994-2020
All Rights Reserved.

When the simulation runs, the savefiles are created, as shown below.

© 1994-2020 All Rights Reserved.

The waveform from the original tstab and the restarted tstab overlay is shown below.

© 1994-2020
All Rights Reserved.

Even in the zoomed-in view, the waveforms overlay, as shown below.

© 1994-2020
All Rights Reserved.

Misc Tab

© 1994-2020
All Rights Reserved.

New Initial Value For Each Point (restart)

Restart is an option that controls the behavior of the simulator in a sweep or in Monte Carlo. If restart is yes, then the simulation restarts from scratch for every sweep point. This can cause excessive runtimes and/or convergence difficulties. When restart is no, the previous member of a sweep or Monte Carlo is used as the starting point for this analysis. If this point does not converge, a variety of methods are applied automatically to try to get convergence. These methods include parameter sweeping, backing off the source power and then ramping it back up in increments, and running a tstab simulation.

xdbsteps

Normally, this option doesn't need to be set. *xdbsteps* is an option that enables you to specify the maximum number steps for the compression point search. The simulator terminates if the value specified in the `xdbsteps` field is exceeded before the compression point in found. The default value is 100.

xdbmax

Normally, this option does not need to be set. The *xdbmax* option enables you to specify the maximum input power or voltage for the compression point search. The default is +30 dBm when *Gain Definition* is set to *Power* and 2.0V when *Gain Definition* is set to *Voltage*.

xdbstart

Normally, this option does not need to be set. The *xdbstart* option enables you to set the starting power or voltage. The default is xdbmax - 70 when *Gain Definition* is set to *Power* and xdbmax/20000 when *Gain Definition* is set to *Voltage*.

xdbtol

Normally, this option does not need to be set. The *xdbtol* option controls the maximum delta between sweep values and sets the maximum delta between the actual harmonic balance last power or voltage level and the largest compression value specified for the compression analysis. The default value for *xdbtol* is 0.01. The actual tolerance between the power or voltage level and the maximum specified power level in the *Choosing Analyses* form for a normal mode compression analysis is +/- (2.5*xdbtol*maximum compression level+0.001). For a *Rapid Mode* compression measurement, the tolerance is doubled. The maximum change in power during a sweep is 10*xdbtol*(xdbmax-xdbstart). The default for *xdbtol* is 0.1, so there will be at least 10 points in the sweep. The starting change in power or voltage is half the maximum change.

© 1994-2020 All Rights Reserved.

file

Use the *file* option to specify the file in which the large-signal S parameters (LSSPs) need to be written.You can specify a file name or click the browse button to browse to the location of the file.

datafmt

The *datafmt* option specifies the format in which LSSP data is output. Possible values are *spectre* and *touchstone*. The default value is *touchstone*.

datatype

The *datatype* option sets the data format for the LSSP output. Possible values are *realimag*, *magphase*, and *dbphase*. The default value is *magphase*.

memory usage estimation

The *memory usage estimation* option provides an estimate of the memory required to perform Harmonic Balance analysis. If this option is set to *yes*, an estimate of the memory required to perform hb analysis is printed in the log file. You can use this estimate to plan the computing resources before performing hb analysis. In memory estimation mode, the simulator performs a short simulation and exits after printing the estimate in the log file without saving the harmonic balance results. You must set this option to *off* to perform actual hb analysis. Memory estimation is not recommended for simulations that require less than 500MB of disk space.

AdditionalParams

This is a field for `<option_name>=value` statements. Multiple `<option_name>=value` statements are allowed with a space between them. Generally, this is used to unlock the beta features, however, there is one case where this field might be useful for an option that is not in the GUI.

In some cases, in the tstab interval, the timestep collapses to near zero. The symptom is that the percent complete stays at the same value for many reporting intervals in the Spectre output window. In this case, there might be a discontinuity in one of the models.

In transient simulation, two things can reduce the timestep; having too much numerical integration error or taking too many iterations at a single timestep can cause the timestep to be cut. Spectre does not report what is causing the timestep to become small. One way to

© 1994-2020 All Rights Reserved.

eliminate the possibility of numerical integration error (which is called local truncation error in the simulator) as the cause of reducing the timestep is to set *lteratio* to 1e9. The disadvantage of this is that it disables the normal method of timestep control, therefore, *maxstep* must be set to maintain accuracy.

Another way to accomplish this is to set *lteminstep* to a value of about the stop time divided by 1e5. This allows the normal method of timestep control, but if for some reason the numerical integration error wants to cut the timestep too small, the error is ignored. *lteminstep* is the way to specify this. To do this in the ADE Explorer, type `lteminstep=<Stop_Time>/1e5`. If you use `lteratio=1e9`, or if you set *lteminstep* and you still have very small timesteps, try increasing *maxiters* in the *Convergence* tab to between 40 and 100. When *lteratio* or *lteminstep* is set and you still have timestep problems, there is likely a discontinuity in either a device model or in a Verilog-A model. An example is shown below.

© 1994-2020 All Rights Reserved.

Oscillator Options

1. *pinnode* can be set manually. Do not start the name with a slash. If you have a hierarchical name, it will usually be in the form *I1.I10.out1*. If you use the probe-based method, do not specify any entry here.

2. The *save osc ppv* option is obsolete.

© 1994-2020
All Rights Reserved.

Key Outputs in the Spectre Window

First, the circuit inventory is displayed. In addition, you also get a message that the linear oscic is being used. This is controlled by the *Calculate initial conditions (ic) automatically* check box.

Next, you get the output that either indicates that the linear oscic worked, and the estimate of the frequency, or that it failed, and is running tstab.

The time of the tstab interval is reported. If *Run transient?* is set to *Decide automatically*, you might see one or more messages that indicate the tstab interval is being extended. This usually is only seen for oscillators.

For oscillators, the tstab interval includes the time set in tstab plus five periods of the frequency estimate from linear oscic. If linear oscic is not used, it includes five periods of the frequency you estimated in the *Choosing Analyses* form.

The list of the options that are actually used for this phase is listed.

© 1994-2020 All Rights Reserved.

The transient output from the tstab interval is next.

© 1994-2020 All Rights Reserved.

Next, one cycle of the oscillator is simulated.

At the end of the one cycle, a Fourier transform is done to serve as the starting point in the frequency domain, and a node is picked for the pinnode.

© 1994-2020 All Rights Reserved.

The frequency domain iterations start.

The iterations continue until the delta norm and the residual norm are both less than one.

© 1994-2020 All Rights Reserved.

The frequency is reported. The times and memory use are reported.

© 1994-2020 All Rights Reserved.

The circuit inventory, pin node, and linear oscic are reported.

The linear oscic frequency is reported.

The options used for this part of the simulation are reported.

© 1994-2020 All Rights Reserved.

The probe-based method inserts a source connected between the pinnodes. The amplitude and frequency are adjusted until the current is near zero.

© 1994-2020 All Rights Reserved.

If the probe-based method is used, the initial starting point is determined. The outer loop iterates on the frequency and the inner loop iterates on the amplitude.

The iterations continue until the source current is near zero, the sum of the currents at all the nodes is near zero, and the delta between iterations is near zero.

The tolerances are set by using the errpreset setting (liberal, moderate, conservative).

© 1994-2020 All Rights Reserved.

The final frequency is reported. The time and memory statistics are shown.

© 1994-2020 All Rights Reserved.

Examples

RF source setup for all examples in this chapter

The *Edit Object Properties* form for the port that is used as the RF source is shown below.

© 1994-2020 All Rights Reserved.

© 1994-2020
All Rights Reserved.

In the *Edit Object Properties* form:

1. If you select *Names* in the *Choosing Analyse*s form, specify a name in the *Frequency name 1* and *Frequency name 2* fields.
2. Set the first sinusoid frequency to a variable name in the *Frequency 1* field.
3. Set the *Amplitude 1 (dBm)* field to a variable name.
4. Set the *Frequency* 2 field to a variable name.
5. Set the *Amplitude 2(dBm)* field to a variable name used to set the first frequency amplitude, if you want the amplitudes to match. If you do not want the amplitudes to match, you can use a different variable name.
6. Select the *Display small signal params* check box.
7. Set the *PAC Magnitude* field to 1 volt.

Note: Using variables in the entries enables you to change the properties without needing to change the schematic. Although this is useful mostly for ADE Explorer, it works in ADE Assembler also. ADE Assembler allows any property in the schematic to be varied directly without changing the schematic.

© 1994-2020 All Rights Reserved.

ADE Explorer Setup for 1-tone Harmonic Balance (hb) Analysis

The Analog Design Environment (ADE) Explorer window is shown below.

Note: The variable name *frf1* is set to `900M` which sets the first frequency in the input port to 900MHz. The variable *prf* sets the input power for the RF input. The second frequency is zero, which disables that input.

To set the 1-tone hb analysis:

1. Open the *Choosing Analyses* form from the ADE Explorer window. To do this, select *Choose* from the *Analysis* menu. Alternatively, expand the *Analysis* option on the Setup pane and click the *Click to add analysis* text or click the *Choose Analysis* button

© 1994-2020 All Rights Reserved.

located on the top right corner of the ADE Explorer window. The *Choosing Analyses* form is displayed, as shown below.

© 1994-2020
All Rights Reserved.

2. In the *Choosing Analyses* form, select *hb*. The *Choosing Analyses* form expands, as shown below.

© 1994-2020
All Rights Reserved.

3. You can specify either the frequencies of the inputs, or get a list of the source names and frequencies in the schematic. *Frequencies* is select by default for hb analysis.

4. *Decide automatically* is the default for *Run Transient?*. This is the recommended setting. Except for the frequency, this will set everything to automatic. If you select *Yes*, you have the option of setting the tstab (transient-aided) time, and the option to automatically manage the tstab stop time.

5. Select *1* to set the number of tones to 1. This is the default.

6. Set the frequencies, number of harmonics, and oversample factor.

   auto calculates the number of harmonics that are needed based on the Fourier Transform at the end of the tstab interval, and is available for the *Decide automatically* and *Yes* selections for *Run transient?*. If the circuit has square wave signals and you are setting harmonics manually, set the number of harmonics to at least the period of the signal divided by the risetime and set the oversample factor to between 2 and 8.

7. Select the *moderate* check box. This is the default setting.

© 1994-2020
All Rights Reserved.

If you need more accuracy, select the *conservative* check box.

8. When you are done with the setup, run the simulation.Spectre will calculate the number of harmonics and display the result in the Spectre output window, as shown below.

© 1994-2020
All Rights Reserved.

When *Names* is selected, all the sources in the circuit are read into the table along with their frequencies. In this case, there is one source with the frequency set to 900M using the variable *frf1*.

To make the changes:

1. Select the line and set the number of harmonics and the oversample factor in the editable fields (shown highlighted above).
2. Click *Change,* select another field, or press the <*Tab*> key.
3. When you are done, run the simulation.

Plotting Spectral Content

1. In the ADE Explorer window, select *Results* - *Direct Plot* - *Main Form*.

© 1994-2020 All Rights Reserved.

The *Direct Plot Form* is displayed, as shown below.

© 1994-2020
All Rights Reserved.

2. Select the *hb* radio button from the *Analysis* section.
3. Select the *Voltage* radio button from the *Function* section.
4. Select the *spectrum* radio button from the *Sweep* section.
5. Select the *rms* radio button.
6. Select the *dB20* radio button from the *Modifier* section.
7. Enter the harmonics in the *Value At (harms)* field to plot the value at specific harmonics. You can enter a list of harmonics separate by a space in this field.
8. If you want this signal to plot automatically when a new simulation is run, select *Add To Outputs*. The expression is automatically evaluated in ADE.
9. Select the node that you want to plot on the schematic.

© 1994-2020 All Rights Reserved.

The output harmonics are shown in the waveform tool. By default, the waveform tool is docked in the ADE Explorer window, as shown below.

The expression is evaluated in the ADE Explorer window, as shown below.

maestro | db_mixer

Name	Type	Details	Value	Plot	Save	Spec
v /out1; hb dB20(V)	expr	db(rms(vh('hb "/out1")))		✓		
v /out1; 3; hb dB20(V)	expr	db(harmonic(rms(vh('hb "/out1")) 3))	-77.04	✓		
v /out1; 1; hb dB20(V)	expr	db(harmonic(rms(vh('hb "/out1")) 1))	-59.8	✓		

© 1994-2020
All Rights Reserved.

You can click the Launch ViVA icon (shown highlighted above) to display the waveform separately in the Virtuoso Visualization and Analysis tool (ViVA), as shown below.

The dB20 modifier is dB with respect to 1 volt, or dBV.

To position the markers, move your cursor near the object you want to read, and type *m*.

Plotting dBm

On the *Direct Plot Form*:

© 1994-2020
All Rights Reserved.

1. Select *hb* from the *Analysis* section.

2. Select the *Power* radio button from the *Function* section.
3. Select the *Net (specify R)* option from the drop-down list.
4. Select *dBm* from the *Modifier* section.
5. Select the appropriate net on the schematic.

© 1994-2020
All Rights Reserved.

6. Click *OK*.

The output spectrum in dBm is displayed.

To position the markers, move your cursor near the thing you want to read, and type *m*.

© 1994-2020
All Rights Reserved.

Sweeps

1. To enable sweeps, select the *Sweep* check box in the *Choosing Analyses* form.

2. Select what you want to sweep from the drop-down list. In the figure above, *variable* is selected.
3. Click *Select Design Variable* and select the variable that you want to sweep.
4. Specify the sweep range in the *Start* and *Stop* fields.
5. If desired, select the *Add Specific Points* check box and specify the additional points.

   Separate the entries with a space.

Plotting Currents

In the past, currents could be plotted from the terminals of the sources in the circuit only. That required adding an iprobe to the circuit in series with the branch. An iprobe is a zero volt voltage source that also allows the current in that branch to be solved.

© 1994-2020
All Rights Reserved.

Starting with the MMSIM 12.1.1 release, the normal selection method in ADE for saving currents is supported. If you want to save the currents, select *Outputs - Save All*, and choose *selected*, *nonlinear*, or *all*. This is shown below.

© 1994-2020 All Rights Reserved.

If you choose *selected*, then also select *Outputs* - *To Be Saved* - *Select On Design*, and make terminal selections on the schematic. Selecting terminals will add expressions under the *Maestro* tab in ADE Explorer to save those selected currents.

© 1994-2020 All Rights Reserved.

To plot the currents, select the *Current* radio button from the *Function* section in the *Direct Plot Form* and click *OK*.

© 1994-2020
All Rights Reserved.

Compression Point

Gain compression enables you to calculate the compression points and compression curves directly, without the need for post-processing or manual setup of power sweeps. Gain compression calculates voltage and power-based compression points and provides the flexibility to choose the gain reference for compression calculations. With gain compression analysis, the simulator outputs the following:

- The power and gain compression curves
- The input-referred power (or voltage) at compression
- The output-referred power (or voltage) at compression

Power Compression and Gain Curves

1. First, set a reasonable number of harmonics. for the simulation to be accurate at the compression point.

© 1994-2020 All Rights Reserved.

2. Select the *Compression* check box.

3. If desired, select *Rapid Mode*. This mode is faster but slightly less accurate, and is suggested for Monte Carlo, sweeps, or corners where a large number of simulations need to be performed. In this mode, you need to supply an estimate of the input-referred compression point.

4. Choose *Power* from the *Gain Definition* drop-down list for compression point calculation.

5. Choose the reference point for gain calculation from the *Gain Reference* drop-down list.

6. Specify the gain compression level in the *Compression Level (dB)* field. This can be a list of compression values separated by spaces in normal mode only. This is shown above in step 3.

7. Specify the output harmonic number in the *Output/Load Harmonic* field. For a two-tone simulation (a mixer), the output harmonic number will be a two-number sequence. Alternatively, you can click *Choose* and select the output harmonic from the table.

8. Specify the instance name of the input source in the *Source* text field. Alternatively, you can click *Select* and choose the source from the schematic. For power gain, the input source must be a port instance.

© 1994-2020 All Rights Reserved.

9. Specify the instance name of the load termination in the *Load* text field. Alternatively, you can click *Select* and choose the source from the schematic. For load termination, the instance can be a port or a resistor.

10. If required, enter a value in the *Backoff* field. When you specify a value (in dB) in this field, there can only be a single value of compression. The simulator runs the compression analysis and then lowers the power from the compression point by the specified value. Next, it runs HB and all the hb<xx> analyses specified at that reduced input power level.

11. Run the simulation.

After the simulation completes, choose *Results - Direct Plot - Main Form*. The *Direct Plot Form* is displayed, as shown below.

In the *Direct Plot Form*:

1. Choose *xdb* from the *Analysis* section.

2. Choose *Power/Voltage/Gain* from the *Function* section.

3. Choose *Power* from the *Format* drop-down list.

4. Click *Plot*.

© 1994-2020 All Rights Reserved.

The power curve is plotted in the waveform tool, as shown below.

5. Now choose *Gain* from the *Format* drop-down list.
6. Click *Plot*.

© 1994-2020
All Rights Reserved.

The gain curve is plotted in the waveform tool. This example shows multiple compression levels of 1, 2, and 5dB.

Input-Referred Compression

In the *Direct Plot Form*:

1. Choose *xdb* from the *Analysis* section.
2. Choose *Compression Point* from the *Function* section.
3. Choose *Input-referred* from the *Compression Point/Curve* drop-down list.

© 1994-2020
All Rights Reserved.

4. If you want to add the expression to the ADE Explorer window so that it plots automatically after a simulation, select *Add To Outputs*.

© 1994-2020 All Rights Reserved.

The input-referred compression levels plot. This example shows 1, 2, and 5dB compression levels.

You can also add *Output-referred* compression in the same way.

© 1994-2020
All Rights Reserved.

Voltage Gain Compression

1. Select the *Compression* check box.

2. If desired, select *Rapid Mode*. This mode is faster but slightly less accurate, and is suggested for Monte Carlo, sweeps, or corners where a large number of simulations need to be performed. In this mode, you need to supply a n estimate of the input-referred compression point.

3. Choose *Voltage* from the *Gain Definition* drop-down box for compression point calculation.

4. Choose the reference point for gain calculation from the *Gain Reference* drop-down list.

5. Specify the gain compression level in the *Compression Level (dB)* field. A list of values separated by spaces is allowed in normal mode. *Rapid Mode* requires a single entry.

6. Specify the output harmonic number in the *Output/Load Harmonic* field. For a two-tone simulation (a mixer), the output harmonic number will be a two-number sequence. Alternatively, you can click the Choose button at the right of the Output/Load Harmonic field and select the output harmonic from the table.

7. Specify the instance name of the input source in the *Source* text field. Alternatively, you can click *Select* and choose the source from the schematic. For voltage gain, the input source can be a vsource or a port.

© 1994-2020 All Rights Reserved.

8. Specify the positive output terminal for voltage gain calculation in the *Output Node +* field. If you do not specify a value then the node is considered as global ground.
9. Specify the negative output terminal for voltage gain calculation in the *Output Node -* field. If you do not specify a value then the node is considered as global ground.
10. If required, enter a value in the *Backoff* field. When you specify a value (in dB) in this field, there can only be a single value of compression. The simulator runs the compression analysis and then lowers the power from the compression point by the specified value. Next, it runs HB and all the hb<xx> analyses specified at that reduced input power level.
11. If data for intermediate nodes in the schematic is desired, then select *Outputs - To Be Saved - Select On Design in ADE Explorer*, and click the nets you want to save. These nets can be plotted in the *Direct Plot Form* after the simulation.
12. Run the simulation.

After the simulation completes, choose *Results - Direct Plot - Main Form*. The *Direct Plot Form* is displayed, as shown below.

In the *Direct Plot Form*:

1. Choose *xdb* from the *Analysis* section.
2. Choose *Power/Voltage/Gain* from the *Function* section.

© 1994-2020 All Rights Reserved.

3. Choose *Voltage* from the *Format* drop-down list.
4. If you want to add the expression in the ADE Explorer window so that it plots automatically after a simulation, select *Add To Outputs*.
5. Click *Plot*.

The Voltage curve is plotted in the waveform tool, as shown below.

1. To plot the data from intermediate nets, select Intermediate Node from the Function Section.
2. Select *Gain*, *Voltage*, or *Phase* from the *Format* drop-down list. *Gain* is shown in this example.

© 1994-2020 All Rights Reserved.

3. Select the harmonic from the Frequency and Harmonics selection.

4. Click the net in the schematic.

© 1994-2020
All Rights Reserved.

5. The data plots without the compression levels shown.

© 1994-2020
All Rights Reserved.

Compression can be combined with sweeps or monte carlo. To run sweeps, enable the sweep button in the *Choosing Analyses* form, and set up a sweep. A temperature sweep is shown below.

© 1994-2020
All Rights Reserved.

Now run the analysis, and either plot the compression points using the Direct Plot Form, or use the ADE outputs added previously to plot the compression point versus temperature.

Loadpull

For a loadpull simulation, use a port for the load. The harmonic port capability is used. The basic idea is to specify the impedance for all the harmonics you do not want in the harmonic

© 1994-2020 All Rights Reserved.

port definition and then loadpull the harmonic you want in the loadpull analysis. An example circuit is shown below.

Normally, the harmonic port is used to define impedances for all the harmonics except for the one you want to loadpull.

1. To set impedances. select the port and type *q*.

© 1994-2020
All Rights Reserved.

2. Select *Harmonic Port* for the *Port mode*.

3. If this has already been done, click the *Display harmonic port set form* check box.

© 1994-2020
All Rights Reserved.

The Edit harmonic port window is displayed, as shown below.

4. Double-click the *<Click to add>* text in the *Harmonic* column.
5. Type *2* in the Harmonic field.
6. Press Enter.
7. Click the *Resistance (Ohms)* column.
8. Type the real part of the impedance.
9. Press Tab.
10. Type in the imaginary part of the impedance.
11. Press Enter.

© 1994-2020
All Rights Reserved.

12. Continue for all the harmonics in the simulation.

13. Click *OK* in the *Edit harmonic port* window.

14. Click *OK* in the *Edit Object Properties* window.

15. Check or check and save the schematic.

© 1994-2020 All Rights Reserved.

16. Add two variables in ADE Explorer. To add a variable, click the *Click to add variable* text under the *Design Variables* section in the *Setup* pane of the ADE Explorer window. The *Editing Design Variables* form is displayed, as shown below.

17. Specify the name of the variable in the name field and the value of the variable in the Value (Expr) field.

18. Click *Add*.

© 1994-2020 All Rights Reserved.

The example above shows `mag` and `theta` variables added. These variables should not be used to set any component values in the circuit.

19. Open the pss or hb *Choosing Analyses* form.

20. If *pss* is selected, use the harmonic balance engine. If you want to use loadpull with PSS Shooting, see Loadpull on page 762.

21. Select the *Loadpull* checkbox.

© 1994-2020
All Rights Reserved.

22. Click the *Select* button on the right of the grayed out *Load Instance* field.

23. In the schematic, select the load port. This example shows a loadpull for the first harmonic. If you want to loadpull the second harmonic, change the harmonic port definition so that the first harmonic has a defined load impedance and the second harmonic is not specified. To loadpull the third harmonic, specify impedances for all the harmonics except the third harmonic in the harmonic port definition.

The instance name is entered in the *Load Instance* field. Direct entry by typing is not allowed for this field.

24. Click the *Variable* button on the right of the *rho (Z)* field. Select the variable you entered in ADE Explorer.

This variable will be used to set the value of the reflection coefficient in the port.

© 1994-2020 All Rights Reserved.

25. Specify a sweep range, and set the spacing.

26. When completed, click *OK*.

27. Click the *Variable* button on the right of the *phi (Z)* field. This sets the angle of the reflection coefficient.

28. In the *Select Design Variable* window, select the second variable you entered and specify the sweep limits.

29. When completed, click *OK*.

The *Choosing Analyses* form is updated.

© 1994-2020
All Rights Reserved.

30. Specify the system resistance in the *Z0* field.

The *Choosing Analyses* form should appear similar to the one below.

© 1994-2020
All Rights Reserved.

31. When completed, click *OK* and run the simulation in ADE Explorer.

32. In ADE Explorer, select *Results - Direct Plot - Main Form*.

There are two methods of plotting loadpull curves.

First Method

1. Select *pss or hb results*.
2. Select *Power Contours*.
3. Select *Magnitude*, *dB10*, or *dBm*.
4. Specify 2 more than the number of contour lines you want.

   The smallest and largest power occurs at a single point which is plotted, but are very difficult to see in the waveform tool.
5. Select the *Close Contours* check box. This causes the contours from the loadpull to be displayed as circles or ellipses, depending on the distortion of the circuit.

© 1994-2020 All Rights Reserved.

6. Select the first harmonic.

© 1994-2020
All Rights Reserved.

7. Select the top terminal of the top port in the triplexer.

   The power contours appear.

Second Method

1. Select *pss* or *hb results*.
2. Select *Power*.
3. Select *variable*.

© 1994-2020
All Rights Reserved.

4. Select *Magnitude*, *dB10*, or *dBm*.
5. Select the *harmonic number*.
6. Select *Loadpull Contour*.
7. Specify 2 more than the number of curves you want.

© 1994-2020 All Rights Reserved.

8. Select the *Closed Contours* check box. This causes the contours from the loadpull to be displayed as circles or ellipses, depending on the distortion of the circuit.

© 1994-2020
All Rights Reserved.

9. Select the top terminal of the top port of the triplexer circuit.

The loadpull curves appear in the waveform tool.

© 1994-2020
All Rights Reserved.

Power-Added Efficiency

1. Set up an hb run as desired. A sweep is shown below.

© 1994-2020
All Rights Reserved.

2. Run the simulation.
3. When the simulation completes, select *Results - Direct Plot - Main Form*. The *Direct Plot Form* is displayed.

© 1994-2020
All Rights Reserved.

4. Select *Power Added Eff*.
5. Set the number of DC sources in the circuit you want to include for the PAE measurement.
6. Select the frequency of the output.
7. Select the top terminal of the load in the schematic.
8. Select the top terminal of the input port.
9. Select the positive terminal of the DC source that supplies power to the amplifier. If you set multiple sources, select the positive terminals of all the sources.

   The PAE is plotted in the waveform tool.

Oscillators

On the *Choosing Analyses* form:

© 1994-2020 All Rights Reserved.

1. Set the *Run Transient?* selection appropriately.

© 1994-2020
All Rights Reserved.

2. For oscillators, selecting the *Detect Steady State* check box is strongly recommended. This will run tstab until steady-state is reached, and then switch to harmonic balance. When *Decide automatically* is set for *Run Transient?*, *Detect Steady State* is always on.
3. Specify a frequency that is within 0.5 times the actual oscillation frequency and 1.5 times this frequency.
4. Set the harmonics and the oversample factor as needed for your oscillator. *auto* harmonics is available when *Run Transient?* is set to *Decide automatically* or *yes*.
5. Always select *conservative*.

   For LC and crystal oscillators, no further action is necessary. For ring oscillators, specify the initial conditions in ADE Explorer to force one stage to the high or low state.
6. If you want to see the startup waveform, select *yes* for saveinit.
7. Select any net with the oscillator signal present for the oscillator node.
8. If you want the output frequency versus tuning voltage, define an appropriate sweep for your oscillator.
9. Run the simulation.

© 1994-2020 All Rights Reserved.

Oscillator Tuning Range

To plot the tuning range, on the *Direct Plot Form*:

1. Click the *Harmonic Frequency* radio button.
2. Select the *Add to Outputs* check box.
3. Select the desired harmonic number to plot from the *Harmonic Frequency* field.
4. Click *Plot*.

© 1994-2020 All Rights Reserved.

The tuning range plot appears.

© 1994-2020
All Rights Reserved.

Plotting Modulation Sensitivity

From plotting the oscillator frequency as shown below, an expression is added in the ADE Explorer window.

1. Double-click the expression under the *Details* column and then click the ... button.

© 1994-2020 All Rights Reserved.

The Calculator starts with the expression in the buffer. This expression will be different for different circuits.

2. Set the plotting mode to *New Subwindow*.

© 1994-2020 All Rights Reserved.

3. Select the *deriv* function.

   The expression changes in the buffer.

4. Click the *Plot* icon.

© 1994-2020 All Rights Reserved.

The modulation sensitivity appears in the new subwindow.

Oscillator Tuning Mode Analysis

The oscillator tuning mode analysis tunes the oscillator so that phase noise can be measured at the desired oscillation frequency. It is supported in both harmonic balance and shooting methods and is typically used in Monte Carlo simulations to obtain the phase noise.

In oscillator tuning mode analysis, the oscillator frequency is treated as the target frequency, and the value of a circuit parameter is obtained to achieve the target frequency. The circuit parameter can be an arbitrary netlist variable or an instance parameter.

1. Open the hb *Choosing Analyses* form.
2. Set up the target frequency in the *Fundamental Frequency* field.
3. Select the *Oscillator* check box.
4. Specify the output node from the `Oscillator node+` field. Alternatively, click *Select* and choose the output node from the schematic. Here, `collector` is selected.

© 1994-2020 All Rights Reserved.

5. Select the *Enable tuning mode analysis* check box.

6. Choose the parameter that needs to be tuned from the drop-down list. Here, *Component Param.* has been selected.

7. Specify the name of the parameter in the *Component Name* field. Alternatively, click *Select Component* and select the parameter from the schematic. When you select a

© 1994-2020 All Rights Reserved.

parameter from the schematic a pop-up window appears that contains a list of selectable parameters.

8. Select a parameter from the list. The *Parameter Name* field is automatically populated with the name of the parameter (*dc* in the figure above).

9. Select the *Tuning range* check box and specify the minimum and maximum ranges for the parameter, if required.

10. Run the simulation.

© 1994-2020
All Rights Reserved.

11. After the simulation completes, click *Results - Direct Plot - Main Form* to display the *Direct Plot Form*.

12. Select *Tuning Parameter* from the *Function* section.

13. Click *Add to Outputs*.

14. Click *OK*.

© 1994-2020
All Rights Reserved.

The value of the tuning parameter is added under the maestro in the ADE Explorer window, as shown below.

Oscillator Tuning Mode With Sweeps

1. Sweeps and Monte Carlo are also supported with tuning mode. Below is an example of a sweep of temperature with tuning mode.

© 1994-2020 All Rights Reserved.

2. Enable sweeps in the hb Choosing Analyses form. In this example, temperature is swept from -40 t0 160C. Tuning mode is enabled as before.

© 1994-2020
All Rights Reserved.

3. Now run the simulation. Because the tuning parameter has already been added to the ADE Explorer window in the previous section, the tuning voltage is plotted after the simulation.

4. The trend in tuning voltage and phase noise is plotted.

© 1994-2020
All Rights Reserved.

Two Input Frequencies

To enable an additional frequency, set the variable in ADE Explorer to the frequency you want instead of 0 (zero). In the window below, the variable frf1 has been set to 1.901G.

To set the 2-tone hb analysis:

© 1994-2020 All Rights Reserved.

1. Open the *Choosing Analyses* form from the Analog Design Environment (ADE) Explorer window.

© 1994-2020
All Rights Reserved.

2. Decide whether you want to set the harmonics manually or automatically for the first tone. For auto tstab, and auto selection of harmonics for the first tone, select *Decide automatically* from the *Run transient?* drop-down list.
3. You can specify either the frequencies of the inputs, or obtain a list of the source names and frequencies in the schematic. *Frequencies* are shown above.
4. Click the appropriate radio button to select the number of inputs.
5. Set the frequencies, number of harmonics, and oversample factor. For an automatic calculation of the number of harmonics for the first tone, set the number of harmonics to auto in Tone1.
6. If you have a frequency divider in your circuit, set the divide ratio. This assumes that the Tone1 input will be divided in frequency.
7. If you want to enable a frequency cut, choose *Select* from the *Harmonics* drop-down list box and set a value for *MaximOrder*.
8. Set the accuracy you want for the analysis.
9. If you want to manually set transient-aided hb, select *Yes* from the *Run Transient?* drop-down list and set a time for the transient tin the *Stop Time (tstab)* field. If you want to see the startup waveform, select *yes* for saveinit.
10. Enable sweeps by checking the *Sweep* checkbox, or use the parametric plot tool in ADE Explorer. In the *Choosing Analyses* form, you can specify up to three variables. In the parametric plot tool, the number of variables is unlimited.
11. Set the variable name to be swept.
12. Set the sweep range and step information.
13. If you want to run specific points, check the appropriate check box at the left side of the *Add Specific Points* field.

© 1994-2020 All Rights Reserved.

The form below shows the hb *Choosing Analyses* form with *Names* selected. Only the differences are shown from the previous section.

© 1994-2020
All Rights Reserved.

14. When *Names* is selected, ADE Explorer reads the schematic and lists all the sources in the circuit.

The *Expr* field lists the entry that sets the frequency, and the values field evaluates the variable expression and lists the value.

Mxham lists the number of harmonics for that tone. *auto* is allowed if *Decide automatically* or *Yes* are selected from the *Run transient?* drop-down list.

Ovsap lists the oversample factor for that tone.

Tstab specifies which tone is eligible for transient-aided hb.

Important

Tstab should be enabled for the tone that causes the most distortion in the circuit.

15. To make changes, select the line and then edit the entries in the fields just below.

16. When you are done with your edits, click *Change,* select another field, or press the <*Tab*> key.

17. When you are done with the setup, run the simulation.

18. When the simulation completes, select *Results - Direct Plot - Main Form* in ADE Explorer.

© 1994-2020 All Rights Reserved.

The form looks as below:

© 1994-2020
All Rights Reserved.

19. The *Plotting Mode* drop-down list box has the following options:

- *Append* - Add the new plot to an existing plot in the selected subwindow. This option is selected above.
- *Replace* - Replace the data in the selected subwindow.
- *New SubWin* - Open a new subwindow and plot the data.
- *New Win* - Open a new window and plot the data.

20. Select the analysis type you want to plot from the list. *hb_mt* is selected in the figure above.

Note: *_mt* means multiple tone.

21. Select the specific item you want to plot from the list in the *Function* section. The figure above shows plotting the voltage on a node.

22. Different items can be plotted by using the *Select* drop-down list. You can choose from a number of different things by clicking the *Select* drop-down list. For example, *Differential nets*.

23. If you want to plot the spectrum at a specific power level, select *spectrum* or *frequency*. If you want to see a swept measurement of a specific harmonic, select *variable*. If you want to plot the time-domain waveform at a specific power level, select *ifft*.

24. For a harmonic plot, select *peak* or *rms*. *rms* is selected in the above figure.

25. Select the data type you want to plot. If you want *dBm*, select *Power* in the *Function* section.

26. Select the desired harmonic or power level (as appropriate for your selection in the sweep section)

27. If you want to add an expression to the ADE window so that it plots automatically after a simulation, select *Add To Outputs*.

At any time when filling the *Direct Plot Form*, appropriate directions on what to do are given at the bottom of the form. In this example, a variable value needs to be selected in the variables table.

© 1994-2020 All Rights Reserved.

This is the wide range display of all the harmonics. *For more information on the waveform tool, see the documentation for ViVA (IC6.1).*

Multiple Divider Mode

Generally speaking, multiple receivers are simulated at the same time to analyze leakage from the LOs into the other radio. An example of what if analysis to find what resistance values (top) and capacitor values (bottom) need to be achieved is shown below. Two LNA and Mixer

© 1994-2020 All Rights Reserved.

chains are shown with resistors coupling the LO of the bottom receiver into the top LNA and capacitors from the top LO into the bottom LNA.

1. In the *Choosing Analyses* form, select *hb*.
2. Select *Yes* from the *Run Transient?* drop-down list, and specify a tstab long enough for the longest divider to cycle at least twice.
3. Select *Frequencies*.
4. Select *Multi-divider Mode*.
5. Select the number of tones applied to the circuit.
6. Type the divider input frequencies in the *Fundamental Frequency* fields.
7. Set harmonics manually. Set enough harmonics to capture at least five harmonics at the input frequency to the divider.

© 1994-2020 All Rights Reserved.

8. Set *Oversample Factor* to two or four. Use four for very square output waveforms.
9. Specify the divider ratio for each signal in the *Freq Divide Ratio* fields.
10. Choose *Select* from the *Harmonics* drop-down list.
11. Select *funnel* cut.
12. Specify 5 in the *MaximOrder (int)* field.
13. Run the simulation.
14. When the simulation completes, select *Results - Direct Plot - Main Form*.
15. Plot the output as desired. In the example below, the complete ifft and the spectral content at the receiver inputs are plotted.

Compression Point

To get the compression point curve on the Direct Plot Form, first run an hb analysis (see Two Input Frequencies).

Follow the steps below to get the compression point curve on the *Direct Plot Form*:

© 1994-2020 All Rights Reserved.

1. Select *Compression Point*.

© 1994-2020
All Rights Reserved.

2. Select the desired value (port, net, or differential net) from the *Select* drop-down list.
3. Select whether you want the output power or the gain plotted on the Y axis as a function of the input power from the *Format* drop-down list.
4. Select the desired amount of compression in the *Gain Compression (dB)* field. For small amounts of compression you might need to use conservative in the hb *Choosing Analyses* form.
5. Specify the desired level of input power to begin drawing the 1dB down ideal 1dB per dB curve. The default is to drop down the amount of compression specified at the lowest power point of the sweep.
6. Select the desired output-referred or input-referred compression point.
7. Select the desired frequency of the output harmonic.

This is the compression plot. The ideal 1dB per dB curve is drawn and the compression point is shown at the upper right of the plot.

© 1994-2020
All Rights Reserved.

IFFT

To get the time-domain waveform, select the following values on the *Direct Plot Form*:

1. Select *Voltage* from the *Function* section.
2. Select *ifft* from the *Sweep* section.
3. For multi tone simulations, the default is to filter out the higher harmonics of the hb analysis in order to improve the plotting speed.
4. Select the desired input power for the ifft plot.

This results in a good waveform that plots quickly. Only the lower frequency harmonics are present in this plot.

© 1994-2020
All Rights Reserved.

To get the best waveform, select the *Complete ifft* checkbox. The plotting time can be considerable for 3 and 4 tone simulations.

Note that not all the harmonics are present in the upper (default) waveform. You choose if you want a quick idea or the plot with all the harmonics. If you want the higher order harmonics in the waveform, select the *Complete ifft* checkbox in the *Direct Plot Form*. The full waveform and the zoomed-in views are shown below.

© 1994-2020 All Rights Reserved.

Compression Vs Pout

Another thing you can plot from a sweep is the compression versus the output power. To do this:

© 1994-2020
All Rights Reserved.

1. Select *Comp Vs Pout* from the *Function* section.

2. Several different types of data can be selected. For example, a net, or differential nets. A port is shown in the figure above.

© 1994-2020 All Rights Reserved.

3. The zero compression input power can be set in the field, as shown below. The default is the lowest power point of the sweep.

4. Select the frequency of the harmonic that you want to plot.

5. Select the port in the circuit.

© 1994-2020 All Rights Reserved.

The output is displayed as below.

Example Using the Harmonic Port with Harmonic Balance

The port component has a feature called the harmonic port that allows the specification of different impedances at different harmonic numbers. This is available for harmonic balance large-signal analyses only, and does not work with any other analysis.

© 1994-2020
All Rights Reserved.

For this example, a 2.4GHz LNA is the example circuit.

© 1994-2020
All Rights Reserved.

The input and output ports are set up normally, as shown in the properties forms below.

© 1994-2020
All Rights Reserved.

PORT1 is the input and PORT2 is the output.

© 1994-2020
All Rights Reserved.

One Input Frequency

The harmonic balance *Choosing Analyses* form has a single input applied.

© 1994-2020
All Rights Reserved.

To use the harmonic port, select *Harmonic Port* on the *Edit Object Properties* form. In this case, the port being used as a load in the circuit is selected. The *Edit harmonic port* window is displayed.

For this setup, a single input frequency is applied to the circuit. In this case, harmonic numbers 1, 2, 3, 4, and 5 exist in the simulation result. In the *Harmonic* column, you enter the harmonic number. In the *Resistance* column, you enter the resistance. In the *Reactance* column, you enter the series reactance in ohms. Positive reactance adds inductance, and negative reactance adds capacitance. Multiple entries are allowed,

© 1994-2020 All Rights Reserved.

therefore, different impedances can be set for each harmonic. The resistance or reactance can be left blank, which is interpreted as zero resistance or reactance.

© 1994-2020
All Rights Reserved.

Here is an example where the third harmonic is set to 1 ohm in series with +j1mohm. This should lower the amplitude of the third harmonic.

The top trace is the normal port, and the bottom is the trace with the third harmonic set to 1ohm in series with +j1milliohm. Note that the third harmonic is smaller.

© 1994-2020
All Rights Reserved.

Two Input Frequencies

Now the harmonic balance *Choosing Analyses* form has two frequencies.

© 1994-2020
All Rights Reserved.

Also, the diamond cut is set with a very large *MaximOrder*. The reason for doing this is to see the harmonic numbers that correspond to the frequencies produced by the circuit. Each frequency has 2 harmonic numbers that indicate which harmonics are mixing with which to produce a specific output frequency. For example, take the 2.44GHz term. The harmonic number is 1 0. The frequency that is produced is 1*2.44G + 0*2.46G. Now look at the intermod at 2.42GHz. This index is 2 -1. The frequency is 2*2.44G - 1*2.46G. These are the harmonic numbers that can be set for the harmonic port. Wild cards (*) are not allowed in the harmonic number specification.

Note that if there were three input frequencies, there would be three numbers that specify every output frequency.

In this case, if the harmonic was specified as 3, there would be an error produced by spectre because there is a single index in the harmonic, but there are 2 indices for all the harmonics of the existing simulation. The number of numbers in the harmonic field must match the number of input freqeuncies in the circuit.

The above example sets the impedance of the third order intermods to be much lower than 50 ohms. If the source is in series, setting the resistance and/or reactance to a large number can block harmonics you do not want. If the port is in parallel like this one, setting small

© 1994-2020 All Rights Reserved.

resistance and reactance values can short out an undesired harmonic. This can be used for debugging the circuit, or for setting values to maximize power-added efficiency.

The top trace above is from the normal port, and the bottom trace is from the harmonic port. The third order harmonics are much smaller because they are essentially shorted out.

Three Tone Harmonic Balance

Generally, a 3 tone harmonic balance simulation is used to calculate large-signal IP3 for transmit applications. If you need a small-signal IP3 measurement, see Rapid IP3 in the hbac section of this chapter. For a small-signal IP3 measurement, Rapid IP3 is much faster and is usually more accurate than a 3-tone harmonic balance simulation.

© 1994-2020 All Rights Reserved.

In a 3-tone simulation, first decide if you want to remove the higher order terms from the solution. If you remove the high order harmonics from the simulation, it can run faster because fewer harmonics are calculated.

© 1994-2020 All Rights Reserved.

On the Choosing Analyses form:

© 1994-2020
All Rights Reserved.

1. To trim the high-order harmonics, choose *Select* from the *Harmonics* drop-down list.

   Funnel is recommended for systems where one or more signals cause a lot of distortion compared to the rest of the signals. Generally, these systems are mixers or sampling circuits that have large amplitude LO or sampling clock signals. *Funnel* calculates the harmonics whose order is less than that specified in *MaximOrder* from the simulation and adds the harmonics of the LO and RF tones but not the higher order mixing products.

   Funnel cuts the harmonics whose order is higher than that specified in *MaximOrder*, but it does not cut the lower-order harmonics of the mixing products. For example, you might set funnel and *MaximOrder* to six. This calculates the sixth order and lower mixing products, but allows the higher order (up to the 10 harmonics specified for Tone1) LO harmonics to be calculated. This is useful for mixer simulations where you know that the LO produces relatively more harmonics than you want with *MaximOrder*. Setting *maximOrder* to six is the recommended starting point for a large-signal IP3 measurement in a down-convert mixer.

2. To see the actual harmonic frequencies to be run in a list, type a frequency range in the fields next to *Harmonics* drop-down list.

3. Click *OK* and start the simulation.

© 1994-2020
All Rights Reserved.

Large-Signal IP3

To plot a large-signal IP3, in the *Direct Plot Form*:

© 1994-2020
All Rights Reserved.

1. Select *IPN Curves*.
2. Select the *Variable Sweep* radio button.
3. Select the input power level you want to do the third order projection from. The default is the lowest power of the sweep.
4. Select Input or Output referred from the drop-down list.
5. Select the order from the *Order* drop-down list. *3rd* order is the default.
6. Select the 2nd or 3rd order frequency in the *2nd Order Harmonic or 3rd Order Harmonic* fields.
7. Select the first order frequency in the *1st Order Harmonic field*.

The IP3 calculation (Input Referred IP3 = 4.97 dBm in the plot below) looks like it is tied to the last datapoint of the third order curve. It is tied there so it will be displayed on the screen. It actually calculates the intercept point.

The extrapolation point (ep = -50 in the plot below) is the input power level that the IP3 calculation is done from. This is the input power level where the ideal curves and the measured data overlay.

© 1994-2020
All Rights Reserved.

In the IP3 plot for a small-signal IP3 measurement, look for the ideal curves and the data to nearly overlay each other on the third order curve for at least a 10dB range of input power. The plot above shows the data and the ideal curves being essentially the same from -50 to about -15 dBm.

The bottom right corner of the plot shows the following in the order as listed below:

- Output of the circuit.
- Third order frequency.
- First order frequency.

Note that the above IP3 is really a small-signal IP3 measurement. For a much faster IP3 calculation use Rapid IP3 as documented in the hbac and pac sections. For a large-signal

© 1994-2020 All Rights Reserved.

IP3 calculation, set the extrapolation point to the desired input power for the large-signal measurement as shown below in the direct plot form.

© 1994-2020
All Rights Reserved.

Now either select replot, or select the output port in the schematic.

Note that the IP3 calculation has changed from 4.97dBm to 2.11dBm.

Multi-Sinusoid hb Analysis

The port component has been enhanced to support the generation of up to nine sinusoids simultaneously. To do this, *Frequency* 2 needs to be set to the spacing frequency. All the sinusoids need to have equal spacing frequencies. *Frequency 1* in the port needs to be set to the center frequency of the multiple output frequencies. The center frequency needs to be one of the frequencies that is generated. For example, imagine that 1GHz, 1.01GHz, and 1.02GHz need to be generated. The center frequency in this case is 1.01GHz and the spacing is 10MHz. Now imagine that 1GHz, 1.01GHz, 1.02GHz, and 1.03GHz need to be generated. Now the center frequency could be either 1.01GHz or 1.02GHz.

Once the center frequency and spacing are defined, click *Display Multi Sinusoid*. Next, specify the number of carriers you want to generate. There is a maximum limit of nine in one port. If you need more than nine tones, add more ports in series, all with the same center frequency and spacing. Specify the frequencies you need to generate in the additional ports.

© 1994-2020 All Rights Reserved.

Divide the port resistance desired by the number of ports you have in series. Note that when you do this, you will need to scale the power in dBm for all the tones in the multi sinusoid because the voltage is set in the port assuming there is a match with the port resistance on the port output. In the case of stacking, the overall resistance is the desired port resistance.

Important

The phase unit for the multi-sinusoid outputs is radians, not degrees.

Once the input frequencies are defined, fields are displayed to specify the frequency and amplitude of each sinusoid to be generated. All the frequencies must be either the center frequency specified in the *Frequency* 1 field, or *Frequency 1* with integer spacing frequency offsets from frequency one.

© 1994-2020 All Rights Reserved.

Below is an example of the port setup to generate three signals.

© 1994-2020
All Rights Reserved.

Note that the frequency 2.45G is not generated for this test. For a composite triple beat measurement, one of the frequencies will be missing. For that frequency, just do not specify it.

When the hb analysis is set up in ADE Explorer for an amplifier, specify only two input frequencies. The center frequency and the spacing frequency should be specified in the *Fundamental Frequency* field.

For harmonics, set the total number of possible carriers in the total frequency range. In this case, there are three signals, but there is a fourth possible harmonic at 2.45GHz. In this case, the number of harmonics needs to be four for both tones.

Since all the inputs are sinusoids, *Oversample Factor* is set to one.

© 1994-2020 All Rights Reserved.

An example is shown below.

© 1994-2020
All Rights Reserved.

Now run the simulation. The output node of the circuit is plotted.

© 1994-2020
All Rights Reserved.

An example with five input frequencies is shown below.

© 1994-2020
All Rights Reserved.

If an LNA-Mixer combination is to be simulated, then there are three tones for this simulation: the center frequency, the spacing frequency, and the LO frequency. For the center and spacing frequencies, five carriers are possible, so five harmonics are set. Ten harmonics are set for the LO. The hb *Choosing Analyses* form is shown below.

© 1994-2020
All Rights Reserved.

Now run the simulation. The down-converted output of the circuit is shown below.

Low Memory Usage For hb Multi-Tone Analysis

When performing multi-tone hb analysis, you can use the *Lowmem* option on the hb Choosing Analysis form to reduce the memory cost. This option is available for all hb and hb-based small signal analyses. Depending on the case, the *Lowmem* option can reduce the peak memory cost by 0 to 70 percent. Memory reductions get larger as the number of total harmonics increases. In general, a 3-tone hb simulation with many harmonics will get the best benefit. Adding small-signal analyses like hbnoise with *Lowmem* to these analyses will also benefit.

Note: Lowmem is supported only for driven multi-tone hb circuits. Autonomous and single-tone hb circuits are not supported.

© 1994-2020 All Rights Reserved.

Lowmem is enabled on the hb Choosing Analysis form when a multi-tone is selected in the *Number of Tones* field:

© 1994-2020
All Rights Reserved.

The following image shows a sample netlist with *Lowmem* set *yes*.

© 1994-2020
All Rights Reserved.

Harmonic Balance AC Analysis (hbAC)

Small-signal Versus Large-Signal Analysis for Conversion Gain

One way to measure conversion gain is to apply the LO and the RF signals in harmonic balance (hb) and calculate the full large-signal response at each RF input frequency, and then plot the output as a function of the sweep frequency. While this works, because all the large-signal effects and many harmonics are calculated, the required time for the simulation can be significant.

The alternative is to use a small-signal approach similar to the AC analysis. The AC analysis runs considerably faster than running the transient at multiple frequencies because it ignores the large-signal effects. In circuits with frequency translation, the AC analysis is unusable because the calculation is based on the DC bias point. In hbac analysis, instead of using the DC bias point, the harmonic balance solution with just the LO is used as the basis of the calculation. The large-signal analysis only calculates the harmonics of a single input, and no mixing products, and then hbac is run after the hb analysis. The overall time savings can be very significant.

The hb solution with the LO calculates the amplitude of the harmonics and it characterizes the nonlinearity of the system. Therefore, in hbac, the output mixing products of the input signal mixing with the LO harmonics can be calculated. Because the signal is considered to be small-signal, harmonics of the hbac test signal are not calculated. Like AC analysis, hbac allows the input to be swept in frequency in order to calculate a frequency response of the output mixing products.

For example, think of a perfectly linear circuit with a single input applied. Because the circuit is perfectly linear, the output spectrum that would be calculated by the hb analysis would have only the first harmonic in it. Because the circuit is linear, if a second input was applied, no frequency translation can occur.

Now imagine a real circuit with a single small input amplitude signal at 1GHz. Because the circuit has nonlinear devices, the output will contain 1GHz and the integer multiples of 1 GHz. Because the input signal is small, the amplitude of the harmonics is small, and if a second input were applied at 1.1 GHz, that signal would be very weakly converted when it is mixed with the harmonics of the circuit. Certainly, 1.1GHz would appear at the output with no mixing. 100MHz and 2.1GHz are produced by mixing with the 1GHz fundamental frequency. 900MHz and 3.1GHz are produced when 1.1GHz mixes with the second harmonic of the 1GHz input. 1.1GHz mixes with the other harmonics as well, producing a wide range of output frequencies from a single input frequency.

Now imagine that the 1GHz signal is large. The output harmonics above the first harmonic are much larger. If the input went up 10dB, the first harmonic goes up 10dB, the second

© 1994-2020 All Rights Reserved.

harmonic goes up 20dB, the third harmonic goes up 30dB, and so on. If a second input at 1.1GHz were applied to the circuit, this mixes with the harmonics of 1GHz and more strongly mixes up and down in frequency.

Therefore, the harmonic content of the hb analysis contains information about how strongly a circuit translates frequencies, and this is used by the small-signal analyses hbac, hbnoise, and hbsp in order to calculate the small-signal conversion gain. This section is about hbac analysis.

Example: Conversion Gain (Down conversion)

Consider the double-balanced diode mixer shown below.

© 1994-2020
All Rights Reserved.

The RF input source has the frequency and amplitude set to variable names in order to allow changing the test setup without needing to change the circuit. This is suggested for all the sources used in the circuit that might need to be changed.

The property list is shown below.

© 1994-2020
All Rights Reserved.

The easiest way to get the variables list into ADE Explorer is to select *Variables -> Copy From Cellview*. To set the values, select the value field to the right of the variable, type the desired value, and press *Enter*.

The LO frequency is 1GHz and its amplitude is 1.4 volts peak. This is +13 dBm. The RF frequency is 1.1GHz, but it is disabled because the amplitude is set to 0 (zero). Setting either the frequency or the amplitude to zero disables the production of that waveform from the port.

© 1994-2020
All Rights Reserved.

The hb analysis needs to be set up first. The example below shows a single input at 1GHz with 51 harmonics. For more information, see the harmonic balance section.

© 1994-2020
All Rights Reserved.

Next, the hbAC analysis needs to be set up. Like AC, the frequency range specified in the form is the input frequency range. *Maximum sideband* should be left blank so that all the mixing products from all the harmonic balance harmonics are present in the output of hbac.

© 1994-2020
All Rights Reserved.

Once the analyses are set up, run the simulation. When the simulation completes, select *Results - Direct Plot - Main Form*.

The example below shows a typical plot.

When the net in the schematic is selected, the waveform is displayed.

© 1994-2020
All Rights Reserved.

Note that all the short line segments are the mixing products that are produced when mixed with the harmonics in the harmonic balance large-signal analysis.

Usually, the frequencies that are produced for the down-converted product are desired. In the *Direct Plot Form*, select *sideband* instead of *spectrum*, and then select the desired

© 1994-2020
All Rights Reserved.

frequency range from the list at the bottom of the form. Only that mixing product will be displayed, as shown below. The conversion loss is about 6dB.

© 1994-2020
All Rights Reserved.

Example: Conversion Gain (Up Conversion)

The process is similar for an up-conversion mixer. In this case the input frequency is low, as shown below.

© 1994-2020
All Rights Reserved.

In the *Direct Plot Form*, both the outputs above and below the LO frequency might be desired. Select the first frequency, and click the second while holding the <*Ctrl*> key. Now select the node in the circuit and the results will plot.

© 1994-2020
All Rights Reserved.

Note that the conversion loss is almost identical for both mixing products.

Overview of Simulation Capabilities

Specialized Analysis=None

Down Conversion: Input Near the First harmonic

At the bottom of the hbac *Choosing Analyses* form, there is a selection called *Specialized Analyses*. The default is *None*, and this is for measuring the conversion gain that is the average conversion gain with the large signal applied to the circuit (usually the LO). The mixing products that are produced are called sidebands. The sideband number is the harmonic number that is being mixed with to provide the output. A positive number means that the output mixing product is higher in frequency than the input frequency specified in the

© 1994-2020 All Rights Reserved.

Choosing Analyses form. A negative number means that the output is lower in frequency than the input frequency. A diagram is shown below.

We tend to think of positive frequencies. The default is to reflect negative frequencies to positive frequencies.

The sideband number is the harmonic that the input mixes with. Negative numbers are downconverted in frequency, and positive numbers are upconverted in frequency.

Again, in hbac, a single input frequency is applied at a time, and then the output mixing products are calculated through the highest harmonic of the harmonic balance simulation specified by the *Maximum sideband* parameter. Leaving that parameter blank is recommended, and this calculates the mixing products for all the harmonics in hb. There is no run-time penalty for calculating all the mixing products. There is a runtime penalty for increasing the number of harmonics in the hb analysis.

Note that the input causes harmonics and the output mixing products cause harmonics. Consider a mixer with a 1GHz LO and a 1.1GHz input. All the LO harmonics are calculated by the hb analysis. When 1.1GHz mixes with 1GHz, an output at 100MHz is produced. Because the circuit is nonlinear, harmonics of 100MHz are produced. hbac is a small-signal analysis, and the assumption is that those harmonics do not matter. They are not calculated. In practical circuits, this limitation is not usually significant because generally, we work with the output frequency itself, not its harmonics.

© 1994-2020 All Rights Reserved.

Up-conversion: Input Near Zero Frequency

In a similar manner, up-conversion is shown below.

Frequency Sweep

This analysis is like AC, where the input frequency is swept. The input frequency range is specified in the hbac *Choosing Analyses* form.

In addition to the frequency, there is a selection called *Sweep Type*. The choices are *relative* and *absolute*. When *absolute* is selected, the frequency range is used directly with no frequency conversion. When *relative* is selected, a *Relative Harmonic* field appears. The frequency sweep is shifted by the specified frequency of the harmonic in the harmonic balance analysis. For example, assume that the harmonic balance is at 1GHz, and a log sweep is desired above the first harmonic of the hb analysis. In this case, you need to select *relative* sweep, and specify 1 in the *Relative Harmonic* field. Next, type 1K to 100M for the frequency range with a log sweep and 3 to 5 points per decade.

© 1994-2020 All Rights Reserved.

The sweep will start at 1G + 1K, and sweep to 1G + 100M using log spacing from 1K to 100M. This is illustrated below.

To use a log sweep below the first harmonic, use the same frequency range, and type -1 as the relative harmonic. This is illustrated below.

© 1994-2020
All Rights Reserved.

Maximum Sideband

Conceptually, an infinite number of mixing products are produced when a single input is applied to a nonlinear system. From the practical point of view, usually only a small number of mixing products need to be measured.

There is a property in the *Choosing Analyses* form called *Maximum sideband* that is used to define how many mixing products should be calculated. If *Maximum sideband* is zero, the hbac analysis will only contain the frequency that appears without mixing or aliasing of any type. This is not equivalent to a linear AC analysis because hbac takes into account the instantaneously varying nature of the LO signal because of the large signal being present in the hb analysis.

When *Maximum sideband* is one, the outputs without frequency translation and the outputs that mix or alias with the first harmonic are present in the hbac analysis. When *Maximum sideband* is 10, the hbac analysis includes mixing through the 10th harmonic of the harmonic balance analysis. Usually the number of harmonics in the hb analysis and the maximum number of sidebands should agree. This is easily accomplished by leaving the *Maximum sideband* field blank in the *Choosing Analyses* form in ADE Explorer.

Setting Harmonics and Sidebands

The process of setting harmonics and sidebands is similar to setting the maximum number of harmonics for the harmonic balance simulation. Start with an estimate based on the harmonic content of the input signal(s) and the nonlinearity of the circuit. Then run a simulation. Raise harmonics and sidebands and run again. If the conversion gain measurement did not change, then the original number of harmonics and sidebands might be able to be reduced. If it did change, then more are needed. Getting a stable hbac output may take more or fewer harmonics and sidebands than the harmonic balance simulation by itself.

Modulated

Modulated hbac is typically used for measuring AM to PM measurements in driven circuits and oscillators. In this case, the input is usually the power supply ripple at low frequency, and the desired output is near the first harmonic of the output. Hbac apples a modulated input signal to the circuit, and is able to measure a modulated output signal from the circuit. The types of modulation are Single Side Band (SSB), AM, and PM.

© 1994-2020 All Rights Reserved.

A diagram is shown below for ssb input.

Note that although the input occurs at one frequency, the AM output can be measured directly from the single input.

© 1994-2020
All Rights Reserved.

When AM is applied at the input, the following diagram is shown.

The difference is that in addition to the input at low positive frequency, another input (to form an AM signal) is applied at low negative input. This second input adds to the output signal near the first harmonic.

© 1994-2020
All Rights Reserved.

For PM, the negative input amplitude is negative, as shown below.

By running all three inputs, the SSB/AM/PM input to SSB/AM/PM output can be calculated. This is a setup choice in the *Choosing Analyse*s form.

In the *Choosing Analyses* form, the input frequency range, and the output harmonic must also be specified.

All the plot functions are implemented in the *Direct Plot Form*.

Rapid IP2/IP3 (AC and PAC)

Note that Rapid IP2 and Rapid IP3 in hbac can only be selected when a single tone hb analysis is run. This would be the LO signal for a mixer. hbAC and AC add the ability to have multiple input frequencies. The Rapid IP2 and Rapid IP3 that is available in the AC analysis form allows the calculation of IP2 and IP3 for amplifiers, or other circuits where there is no frequency translation. In hbac, Rapid IP3 and Rapid IP2 are provided for circuits that translate frequencies, whether that translation is an up or down conversion. Note that hbac and ac are small-signal analyses, so the limitation is that the measurement is a small-signal IP3. If you need a large-signal IP2 or IP3 measurement, use a large-signal analysis like hb or qpss to make the measurement. Note that when you are using Rapid IP2 or Rapid IP3 in the AC analysis, either a SpectreRF license is required, or an additional mmsim token is required to run the analysis.

© 1994-2020 All Rights Reserved.

hbAC and AC Rapid IP3 also add the ability to calculate the harmonics and mixing products that are generated by the two input frequencies. Make sure that the input level is in the small-signal range. Rapid IP2 and IP3 have exceptionally small numerical noise floors, therefore, it is recommended that you specify a level that is too small, rather than too large.

The plot functions for Rapid IP2 and IP3 are implemented in the *Direct Plot Form*. Select *Results - Direct Plot - Main Form*. In the *Direct Plot Form*, select *ac* or *hbac* from the *Analysis* section. Next, select *Rapid IP2* or *Rapid IP3* (depending on which was set up) from the *Function* section, and click *Plot*. The X and Y points are shown for the intercept. The X value is input-referred IP2/IP3, and the Y is output-referred IP2/IP3.

Triple Beat

Triple Beat is similar to Rapid IP3, except three inputs are allowed. This is provided for the case where there are two small-signal tones closely spaced in frequency from the transmitter that leak in to the receiver along with a small-signal intended RF input signal. In this case, a third order product is produced at the transmitter tone spacing frequency away from the RF input frequency. Note that triple beat is a small-signal analysis. The amplitude of all the tones must be in the small-signal region, or the result will be incorrect.

Compression and IP2 distortion summary

The distortion summaries measure distortion in an amplifier in AC or in circuits that have frequency translation in hbac. The idea is to find which devices in a signal path contribute relatively more or less to the distortion. Since IP3 and compression are related mathematically, the compression distortion summary provides information about which devices contribute the most amount of third order intermodulation distortion. The IP2 distortion summary measures which devices contribute to the second order distortion. Note that when you are using the distortion summaries in the AC analysis, either a SpectreRF license is required, or an additional mmsim token is required to run the analysis.

Both distortion summaries use the *PAC Magnitude* property in the input port. Make sure that the PAC Magnitude (in volts or dBm) is actually in the small-signal range, or the results will not be accurate. This is because the hbac results do not take into account the large-signal limits like slew-rate or power supply voltage.

The distortion summaries provide a list output, not a plot output. Accordingly, instead of selecting *Results - Direct Plot - Main Form*, select *Results - Print - HBAC Distortion Summary*.

For the compression distortion, you get the actual gain with the distortion of each device by itself divided by the ideal small-signal gain provided by the hbac analysis. In addition, the amplitude of the first three harmonics of the linear output frequency is calculated.

© 1994-2020 All Rights Reserved.

For the IM2 distortion summary, you get the total IM2 output amplitude, and the amplitude that results from having only a single device in the circuit.

Sampled

Sampled is provided in order to calculate the instantaneous conversion gain at a voltage threshold crossing in the ifft or at specific timepoints in the ifft. This is useful for measuring the conversion gain from the power supply to the oscillator output at a threshold crossing, or at specific times in the LO waveform to the output of a mixer.

© 1994-2020 All Rights Reserved.

ADE Explorer Implementation

Circuit, Input port Setting, and ADE Explorer Setup for all the Examples in This Section

Consider the following circuit:

This above circuit is an LNA and a mixer for a 2.4GHz application that is implemented in CMOS. The input source (a port in Cadence terminology) has the values for its frequency and amplitude set to variable names. The PAC magnitude property is set to 1 volt in order to allow easy conversion gain measurements. The reason for the variables is that the frequency and amplitude can be changed in the ADE Explorer environment without changing the schematic.

© 1994-2020 All Rights Reserved.

The *Edit Object Properties* form for the input port is shown below.

© 1994-2020
All Rights Reserved.

ADE Explorer has the variables set to appropriate values, as shown below.

Note that the variables *fb1* and *fb2* (which set the input frequency of the input port) are set to zero, which effectively disables the RF input. Only the LO signal is applied to the circuit.

© 1994-2020 All Rights Reserved.

Using hbAC Analysis for Conversion Gain Measurement

Set up the hb analysis with just the large signal that causes the frequency translation to occur. For more information, see the harmonic balance section.

© 1994-2020
All Rights Reserved.

Set up the hbac Choosing Analyses form as follows:

1. Specify the input frequency range.
2. For hbac, or pac where pss is set to the Harmonic Balance engine, the *Maximum Sideband* field should be left blank in which case, all the mixing products are calculated for all the harmonics in the hb analysis.

© 1994-2020 All Rights Reserved.

3. For a normal conversion gain measurement, set the *Specialized Analysis* field to *None*.

   For a conversion gain measurement, set the PAC magnitude to 1 volt on the *Edit Object Properties* form for the RF input port as shown earlier. This is always volts peak.

© 1994-2020
All Rights Reserved.

4. Now run the simulation. When the simulation completes, select *Results - Direct Plot - Main Form*.

   The *Direct Plot Form* is displayed.

The harmonic balance analysis characterizes the nonlinearity of the system as a series of harmonics. Hbac injects an input frequency, and because the system nonlinearity has been characterized by the harmonic balance simulation, the input tone mixes with the harmonics of the LO to produce outputs at multiple frequencies. These mixing products are calculated for mixing the hbAC input frequency range with the hb harmonics. Because the amplitude of the RF input signal is assumed to be small, the harmonics of the hbac input and harmonics of the output mixing products are not calculated.

To measure the outputs you desire, place a marker at the desired output frequency of interest and read the conversion gain, as shown below.

© 1994-2020 All Rights Reserved.

The low frequency conversion gain is just over 19dB.

In the *Direct Plot Form*:

© 1994-2020
All Rights Reserved.

1. To plot just the desired output mixing product, select *sideband* instead of *spectrum*, and select the desired output mixing product form the list, as shown below.

© 1994-2020
All Rights Reserved.

2. Now select the output node, or click the *Replot* button in the *Direct Plot Form*. The waveform tool displays the output, as shown below.

3. Position a marker by moving the cursor to the desired output frequency and type *m*. The marker appears, as shown in the figure above.

4. To position the marker at an exact frequency, place it as above, and then move your cursor over the intercept point on the trace. Click and hold the right mouse button, move

© 1994-2020
All Rights Reserved.

to *Marker Properties*, and release the mouse button. The *Point Marker Properties* window appears, as shown below.

5. Type the frequency in the *Position* field, and click *OK* or *Apply*.

© 1994-2020
All Rights Reserved.

Commonly Used HBAC Options

© 1994-2020
All Rights Reserved.

Freqaxis

freqaxis specifies whether you want to see the negative frequency axis or not.

© 1994-2020
All Rights Reserved.

The analysis calculates the output frequencies and amplitudes, so *out* and *absout* are reasonable choices. *out* displays the negative frequency axis. *absout* (absolute value of the output) displays positive output frequencies. This is the default. Below is the default of *absout*.

© 1994-2020
All Rights Reserved.

Next is the output with freqaxis set to *out*. This is the same data as the above plot except the negative output frequencies have not been reflected up to the positive frequencies.

If you select *in*, all the outputs at the different frequencies are plotted on the same input frequency range scale. This is not recommended. The default is *absout*.

Tolerance

Leave this option at the default value.

Hbac uses an iterative solver to calculate the output amplitudes. Any iterative solver needs an error tolerance to specify when to stop iterating because the solution is accurate enough, and the tolerance option specifies that accuracy for hbac. For driven circuits, the default is 1e-6. For oscillators, the default is 1e-4.

Resgmrescycle

Leave this option at the default.

© 1994-2020 All Rights Reserved.

Hbprecond_solver

This option has two settings; basicsolver and autoset. Basicsolver is the only solver available in standard Spectre. Autoset is the default solver when APS is used. This solver is faster. If stagnation is detected, Spectre prints a message in the Spectre output window.

Annotate

This option controls the amount of output in the spectre output window. Selecting title produces the least amount of detail, and selecting steps gives the most amount of detail. The default is status.

Save

The default is *allpub*. This saves all the public voltages at all levels of the hierarchy in the schematic. It excludes the internal nodes of the device models. *all* adds the internal nodes of all the devices. *lvl* saves all the nodes including the internal nodes of the devices through the level of hierarchy set in the *nestlvl* option. *lvlpub* is like *lvl*, but it does not save the internal nodes. *selected* saves only the nodes that are specifically saved. In ADE Explorer, this is accomplished using the *Outputs - To Be Saved - Select On Design* and then selecting the nodes and terminals specifically in the schematic. At the netlist level, this is accomplished by using a save statement with a list of names to be saved.

Nestlvl

If *save* is set to *lvl* or *lvlpub*, this option controls the maximum level of hierarchy to be saved. If *nestlvl* is *1*, only the top level is saved. If *nestlvl* is 2, the top level and the next level down are saved. The value for *nestlvl* can be any integer. The example below shows *lvlpub* selected that saves two levels of hierarchy.

AdditionalParams

additionalParams is typically used for new features that are being beta tested early in the release cycle where there is not a gui selection available yet.

© 1994-2020 All Rights Reserved.

hbAC With Multiple hb Inputs

Harmonic balance is available in the hb form and the pss and qpss forms when harmonic balance is selected for the engine. Hbac is available in the hbac form and in the pac and qpac forms when the engine is set to harmonic balance in pss and qpss. In the example below, hb and hbac were used. The LO is set to 2.4GHz as before. To set up the second (RF) input for the harmonic balance analysis, the variable `fb1` was set to `2.5G` in the ADE Explorer window, as shown below. This variable sets the frequency of the first source in the RF port component. In this case, the LO is at 2.4GHz and the RF is at 2.5GHz.

© 1994-2020 All Rights Reserved.

The harmonic balance analysis was set up to calculate the LO harmonics automatically, and three RF harmonics. For more information on Harmonic balance, please see the Harmonic Balance section at the beginning of this chapter.

© 1994-2020
All Rights Reserved.

The hbac form is set up exactly the same as the hbac form was for the first example in the ADE Explorer implementation section.

© 1994-2020
All Rights Reserved.

Now the analysis was run. The harmonic balance output plot below shows that because there were more input frequencies, many more harmonics are produced by the circuit, which is as expected.

© 1994-2020
All Rights Reserved.

Because there are many more harmonics to mix with, many more mixing products are produced by hbac, as shown below.

The mixing products below -200dB are at the numerical noise floor of the simulation.

© 1994-2020
All Rights Reserved.

The next figure shows a zoomed in area near zero frequency. Only the mixing products larger than -25 dB are shown.

If the output frequency were 6GHz, note that there are three different values. Which one is correct? Actually, they all are. They are produced by mixing with different harmonics from the large-signal analysis.

© 1994-2020
All Rights Reserved.

To identify which harmonics are being mixed with, select the top trace. On the left side of the display an entry is highlighted. The example below shows the top trace selected. (It is bold. It is difficult to see in the figure below, but in the waveform tool it is much easier.)

On the right side of the legend, it reads (-1 0) The first number is the harmonic number of the tone with tstab set, or is listed as *Tone1* in the hb *Choosing Analyses* form. This is LO in this example. The second number is the RF input.

-1 0 means that the input frequency is mixed down with the first harmonic of the LO, and not mixed with the RF. The input frequency range in the hbac (qpac) *Choosing Analyses* form is 2.40001G to 2.8G. If you take 2.40001G as a second small-signal input (that is what hbac does) and mix it down with the first harmonic of the LO at 2.4G, you get very near zero. When you mix 2.8G down with 2.4G, you get 400M. So this component starts at near zero, and sweeps up to 400M as the input sweeps from 2.40001G to 2.8G.

© 1994-2020 All Rights Reserved.

Now select the second trace down, as shown below.

The legend reads 1 -2.

2.40001G mixes up with the first harmonic of the LO, and then mixes down with the second harmonic of the RF. So, 2.40001G + 2.4G - 2*2.5G is very near -200M. As the input sweeps up in frequency, the output becomes less negative, passes through zero, and then sweeps up to +200M. Because the negative frequencies are reflected to the positive frequency domain, the resultant response looks like a less than sign (<) in the above plot.

© 1994-2020 All Rights Reserved.

Note that the ambiguity goes away when the frequency sweep on the input source is restricted to a smaller range.

© 1994-2020
All Rights Reserved.

Rapid IP3/IP2

Note: The first discussion is about Rapid IP3.

Rapid IP3 and Rapid IP2 are available in the AC and the hbac *Choosing Analyses* forms. Rapid IP3 in the AC form is for measuring IP3 of amplifiers where there is no frequency translation, and Rapid IP3 in the hbac form allows the measurement of IP3 where frequency translations are present. Rapid IP3 and IP2 measure small-signal IP3 and IP2.

© 1994-2020
All Rights Reserved.

1. First set up the hb form with just the LO applied, as shown below. For more information, see the harmonic balance section.

© 1994-2020
All Rights Reserved.

2. Select the hbac analysis. At the bottom of the form, in the *Specialized Analysis* section, select *Rapid IP3* or *Rapid IP2*, depending on what you want to measure. The entries for the forms are identical, except for the frequency of the intermodulation product. For switched FET mixers, make sure that the psp model is used. Incorrect results will be produced for BSIM3 and BSIM4 models due to a limitation in these models.

© 1994-2020 All Rights Reserved.

3. Define the input port, input frequencies, input power, the frequencies of the linear and second or third order products, and the output node. An example setup for Rapid IP3 is shown below. Make sure the input power is in the small-signal range for your circuit.

© 1994-2020
All Rights Reserved.

4. Leave the *Maximum Non-linear Harmonics* field blank.
5. Run the simulation. When the simulation is complete, select *Results - Direct Plot - Main Form*.

6. Select *hbac* results.
7. Select *Rapid IP3* or *RapidIP2*, whichever was run.
8. Click *Plot*.

© 1994-2020
All Rights Reserved.

The waveform is displayed. The marker shown at the intercept point shows the input and output-referred IP3 or IP2.

Triple Beat

Triple beat is similar to Rapid IP3 and IP2 except it allows three inputs instead of two. Currently, it is available only in the hbac *Choosing Analyses* form. This capability is provided for the case where two small-signal closely spaced inputs from a transmitter leak into a receiver along with the standard RF input to the receiver. The use model is similar to Rapid IP3. Apply the LO signal in hb, and then use hbac triple beat to simulate the three inputs to the receiver. This is a small-signal analysis. If the input levels for all three tones are not in the small-signal range, the result will be incorrect because hbac ignores large signal effects.

© 1994-2020 All Rights Reserved.

1. To run triple beat, first set up the hb *Choosing Analyses* form with just the LO signal applied.

© 1994-2020
All Rights Reserved.

2. Set up the hbac form with the RF signal and the two leakage signals. In this example, the RF signal is at 2.405G, and the other two signals are 1MHz apart. With the LO at 2.4GHz, this gives an intended baseband output at 5M, and triple beat products at the transmitter spacing (1MHz) above and below the 5MHz output.

© 1994-2020
All Rights Reserved.

3. Run the simulation,
4. Choose *Results - Direct Plot - Main Form* to open the *Direct Plot Form*.
5. Select *hbac* from the *Analysis* section.
6. Select *Triple Beat* from the *Function* section.
7. If desired, select *Add To Outputs*. This will add an expression under the *Maestro* tab of the ADE Explorer window.

© 1994-2020
All Rights Reserved.

The waveform tool plots the linear ratio of IM3 divided by IM1 with the X axis values from 1 to 10. This function is more useful in the ADE Explorer window itself.

© 1994-2020
All Rights Reserved.

8. In ADE Explorer, click the *Plot Outputs* icon. The expression updates with the linear ratio of IM1 divided by IM3

.

9. To convert the output to dB, double-click the expression in the area highlighted by the blue box above and click the Open Calculator button (shown with three dots).

© 1994-2020 All Rights Reserved.

10. The calculator opens with the expression in the buffer.

11. If necessary, select *All* just below the *Function* Panel legend. This is highlighted above. Select the *dB20* function.

© 1994-2020 All Rights Reserved.

12. The expression changes, as shown below.

13. Click *Tools - Send Buffer to ADE Outputs*.The expression is added under the Maestro tab of the ADE Explorer window, as shown below.

© 1994-2020 All Rights Reserved.

14. Double-click the empty cell under the Name column and add a title, such as Triple_beat_dbm.

© 1994-2020 All Rights Reserved.

15. In ADE Explorer, click the *Plot Outputs* icon. The ratio is now in dB.

© 1994-2020 All Rights Reserved.

16. If the IM1 and IM3 levels are desired, choose *Tools - Results Browser*.

© 1994-2020 All Rights Reserved.

The Virtuoso Visualization and Analysis window is displayed.

17. Double-click the *hbac_mb* results.

© 1994-2020
All Rights Reserved.

18. Click the right mouse button on *IM1 output V(V)=<`your result_here`>* expression and select *Calculator*. The calculator opens with the expression in the buffer.

© 1994-2020 All Rights Reserved.

19. In the calculator, examine the expression. If there is a slash before *IM1*, remove it. Refer to the red box above. If desired, add the *dB20* function.

20. Click *Tools - Send buffer to ADE Outputs*.

21. If the level for the IM3 product is desired, follow the previous steps, but send the IM3 expression to the calculator.

22. Specify a name for the expressions under the *Name* column of the ADE Explorer window.

© 1994-2020 All Rights Reserved.

23. When you are done, click the *Plot Outputs* icon in ADE. The results are displayed in Value column, as shown below.

Compression Distortion Summary

Note that the *Compression Distortion Summary* is available in the AC and the hbac *Choosing Analyses* forms. *Compression Distortion Summary* in the AC form is for measuring Compression Distortion of amplifiers where there is no frequency translation, and *Compression Distortion Summary* in the hbac form allows the measurement of Compression Distortion where frequency translations are present.

With the Compression Distortion summary, you can see which components in a frequency translating circuit cause relatively more and less distortion.

Note that compression and IP3 are related mathematically, so this analysis gives information about which components contribute more and less to the IP3 measurement as well as which components contribute to compression.

© 1994-2020 All Rights Reserved.

1. First set up the hb analysis with just the LO applied, as shown below. For more information, see the harmonic balance section.

© 1994-2020
All Rights Reserved.

2. Now select the hbac analysis.
3. Select *Compression Distortion Summary* from the *Specialized Analyses* drop-down list.

   The following notice appears to remind you to set the PAC amplitude parameter in the source in the small-signal range.

4. If you leave the *Contributor Instances* field blank, all the nonlinear devices in your circuit will be run. If you want a subset, click *Select*, and select the instances that you want on the schematic.
5. Specify the input frequency in the *Frequency of Input Source* field.
6. Specify the output frequency in the *Frequency of Linear Output Signal* field.
7. Leave the *Maximum Non-Linear Harmonics* field blank.
8. Select the output node in the circuit.

© 1994-2020 All Rights Reserved.

The completed form should appear similar to the one shown below.

© 1994-2020
All Rights Reserved.

9. In the *Edit Object Properties* form, make sure that the *PAC magnitude (dBm)* is set in the small-signal range. In this example, it is set to `-30Bm`.

10. If you make changes to your schematic that you want for the distortion summary, but you do not want to save them to the circuit file, make the desired changes and then select *Check - Current Cellview* in the schematic window. Later, if you want to save them, just

© 1994-2020 All Rights Reserved.

select *Check* and *Save*. If you do not want them, quit the schematic window, and do not save the changes.

ADE Explorer will netlist from the checked view, so it will get the changes.

11. When you are done, run the simulation.

12. When the simulation completes, select *Results - Print - hbAC Distortion Summary*.

© 1994-2020 All Rights Reserved.

The summary is displayed below:

- The first column is the instance name in the schematic.
- The second column is the gain with the signal applied divided by the ideal gain.

  Gain compression is shown as a negative number. Gain expansion is shown as a positive number.
- The third, fourth, and fifth columns are the amplitudes of the first, second, and third harmonic of the linear output frequency with just the nonlinearity of that single device taken into account.

IM2 Distortion Summary

The steps for the IM2 distortion summary are very similar to the Compression distortion summary in the previous section. The differences are noted below.

- The IM2 distortion summary provides the output voltage (magnitude, real, and imaginary parts) of the 2nd order distortion signal for each component in the schematic by itself.
- The IM2 distortion summary does not provide a selection mechanism for the devices in the circuit. All the devices are run in the IM2 distortion summary.
- The output format is slightly different. The first column is the instance name in the schematic. The second column is the magnitude of the IM2 product that is produced by

© 1994-2020 All Rights Reserved.

that single component in volts. The third column has the real and imaginary parts of the voltage. A sample is shown below.

```
HBAC IM2 Distortion Summary

Results in        hbac_im2distortion

Instance          Distortion(V)[Magnitude Complex]
Total             401.43f    complex(-400.233f, -30.973f)
/I33/I1/I0/PM5    1.3848m    complex(-1.3848m, -2.83457u)
/I33/I1/I0/PM3    1.3848m    complex(1.3848m, 2.83457u)
/I33/I1/I0/PM4    1.38479m    complex(1.38479m, 2.83456u)
/I33/I1/I0/PM2    1.38479m    complex(-1.38479m, -2.83456u)
/I33/I1/I1/PM2    1.37327m    complex(-1.37327m, -2.82163u)
/I33/I1/I1/PM4    1.37327m    complex(1.37327m, 2.82163u)
/I33/I1/I1/PM3    1.37327m    complex(1.37327m, 2.82162u)
/I33/I1/I1/PM5    1.37327m    complex(-1.37327m, -2.82162u)
/I33/I1/I1/I2/NM33    154.322n    complex(113.004n, -105.097n)
/I33/I1/I1/I2/NM3    154.322n    complex(-113.003n, 105.097n)
/I33/I1/I0/I2/NM3    150.693n    complex(-111.65n, 101.206n)
/I33/I1/I0/I2/NM33    150.693n    complex(111.649n, -101.207n)
/I33/I1/I1/I2/PM12    9.1563n    complex(7.15786n, -5.70989n)
/I33/I1/I1/I2/PM11    9.15628n    complex(-7.15786n, 5.70986n)
/I33/I1/I0/I2/PM11    9.12122n    complex(-7.28648n, 5.48669n)
/I33/I1/I0/I2/PM12    9.12118n    complex(7.28648n, -5.48664n)
/I33/I1/I1/NM0    4.42456n    complex(-4.42455n, -9.36531p)
/I33/I1/I1/NM1    4.42456n    complex(4.42455n, 9.36527p)
/I33/I1/I0/NM0    4.22614n    complex(4.22614n, 8.5599p)
/I33/I1/I0/NM1    4.22614n    complex(-4.22613n, -8.55975p)
/I33/I1/I1/PM1    2.66145n    complex(-2.66145n, -5.45575p)
/I33/I1/I1/PM0    2.66143n    complex(2.66142n, 5.4525p)
/I33/I1/I0/PM1    1.24769n    complex(1.24769n, 2.75393p)
/I33/I1/I0/PM0    1.24766n    complex(-1.24765n, -2.74832p)
/I33/I0/NM9       358.96p    complex(-358.957p, -1.40184p)
/I33/I0/NM11      358.957p    complex(358.954p, 1.40182p)
/I33/I1/I1/I2/NM2    178.011p    complex(-160.898p, 76.1576p)
/I33/I1/I1/I2/NM1    178.011p    complex(160.898p, -76.1575p)
/I33/I1/I0/I2/NM2    101.771p    complex(67.2487p, 76.3869p)
/I33/I1/I0/I2/NM1    101.771p    complex(-67.2486p, -76.387p)
/I33/I1/I0/I2/NM32    83.8968p    complex(83.8604p, 2.47293p)
/I33/I1/I0/I2/NM0    83.8968p    complex(-83.8603p, -2.47289p)
/I33/I1/I1/I2/NM0    55.5677p    complex(-55.5163p, -2.3902p)
/I33/I1/I1/I2/NM32    55.5677p    complex(55.5163p, 2.39017p)
/I33/I0/NM12      22.2028p    complex(22.2027p, 53.3724f)
/I33/I0/NM10      22.195p    complex(-22.1949p, -53.3496f)
```


© 1994-2020 All Rights Reserved.

Modulated hbAC Analysis for an Oscillator

Modulated hbac analysis is usually applied to oscillators to measure AM to PM conversion from low-frequencies on the power supply (ripple) to the output near the first harmonic. It can also be applied to digital circuits for the same purpose. Although the shape of the transfer function is different, the steps are the same for both applications. Note that a relatively large number of harmonics and an oversample factor of at least 4 is needed for simulating the digital circuit.

In the circuit, make sure that any PAC magnitude terms are removed on the input source (if there is one), and set the PAC Magnitude to 1 volt in the source that is used as the power supply. If you do not want to save the change permanently, just select *Check - Current View* in the schematic.

This example is for an oscillator. The circuit is shown below:

© 1994-2020
All Rights Reserved.

In ADE Explorer, set the values for the power supply voltage and for the tuning voltage.

© 1994-2020 All Rights Reserved.

Set up the harmonic balance analysis For more information, see the Harmonic Balance section at the beginning of this chapter.

© 1994-2020
All Rights Reserved.

In the hbac *Choosing Analyses* form, first select *Modulated* in the *Specialized Analyses* section, as shown below.

Note that this causes the *Sweep Type* to be set to *relative*. Since the gain from power supply ripple to the output is desired, the relative harmonic number is zero, and then the desired

© 1994-2020
All Rights Reserved.

frequency range is entered. *Maximum sideband* can be left blank, or it can match the hb harmonics, as shown above.

Set the *Input type* to *SSB/AM/PM*.

Next, the output harmonics can be set. Note that a list of harmonics can be entered. The easiest way is to click the *Choose* button to the right of the field and select the desired frequencies from a list in the *Choose Harmonic* window, as shown below.

The same process is used to set the input frequency, but only one frequency range is allowed.

Once the analysis has been set up, run the simulation. When the simulation completes, select *Results - Direct Plot - Main Form*.

© 1994-2020 All Rights Reserved.

In the Direct Plot form select *SSB* for *Input*. Select *PM* for *Output*. Since many output frequencies were calculated, you need to select the one you want to plot. In this example, the output just above the first harmonic is selected. Once this is done, select a net in the schematic.

© 1994-2020
All Rights Reserved.

The output is displayed. To make a measurement at a specific frequency, position a marker. The measurement is in radians per volt of input on the supply that has PAC magnitude set.

© 1994-2020
All Rights Reserved.

Sampled hbAC Analysis for a Mixer

hbAC sampled is for measuring the conversion gain at a specific instantaneous time in the ifft waveform. The first step is to determine an appropriate threshold for the measurement by setting up the hb analysis, as shown below.

© 1994-2020
All Rights Reserved.

Now run the analysis. When the analysis completes, select *Results - Direct Plot - Main Form* in ADE Explorer and plot the time-domain waveform, as shown below.

© 1994-2020
All Rights Reserved.

The waveform is shown below.

Note that 2.5 volts is a good threshold voltage.

Now set up the hbac *Choosing Analyses* form.

© 1994-2020
All Rights Reserved.

© 1994-2020
All Rights Reserved.

At the top of the form, set up the input frequency range like normal.

1. At the bottom of the form, select *Sampled*. Usually the gain to a node is desired, therefore, select *voltage*.
2. Click *Select* located to the right of the *Net+* field and select the node in the schematic.
3. Specify the desired threshold voltage based on the ifft from the hb analysis
4. Choose an option from the *Crossing Direction* drop-down list. In the example, *all* is selected.
5. If you need more than 16 crossing points, type the number you need in the *Maximum Samples* field.
6. If you want to sample at specific times from the ifft waveform, you can type in the times in the *Additional Timepoints* field.

© 1994-2020 All Rights Reserved.

7. Now run the simulation. When the simulation completes, open the *Direct Plot Form* and set it up, as shown below.

8. Select *hbac* results.

9. Select *Voltage*.

© 1994-2020
All Rights Reserved.

10. Select *spectrum*.

11. Select *dB20*.

12. Select the time you want from the *Variable Value (eventtime)* list.

13. Select the output node in the schematic.

The waveform is plotted, as shown below.

Because the sweep was log spaced, the X Axis of the waveform tool is set to log. To zoom in to the frequency of interest, double-click one of the numbers on the X Axis, and type in the output frequency range of interest. This is shown below.

© 1994-2020
All Rights Reserved.

To measure the conversion gain to the output, move the tracking cursor to the output frequency and read the gain.

© 1994-2020
All Rights Reserved.

Harmonic Balance Noise Analysis (hbnoise)

Small-Signal Versus Large-Signal Analysis for Noise

In most cases, noise is a small-signal problem in the real world. Consider an amplifier. One approach is to calculate the small-signal gain from the noise sources to the output, then calculate the frequency response of the noise source, and then add up the noise power at the output. This is done in the linear noise analysis. This runs quickly because the gain calculation just takes the tangent at the DC operating point for everything in the circuit. This ignores large-signal effects like slew-rate limits and clipping. Because noise in most circuits is actually very small compared to the output levels, this approach works well. Because it is a linear approach, frequency translations cannot be calculated.

When the input to the amplifier gets large enough that it starts to create nonlinearity, the frequency translations in the circuit need to be taken into account. Noise near the harmonics can mix up or down to the output frequency. In this case, the large signal needs to be applied in a large-signal analysis like harmonic balance (hb) where the nonlinearity and harmonic levels can be calculated. After the hb runs, hbnoise runs and is similar to linear noise except the frequency translations of the circuit are taken into account. Instead of calculating a single gain from the noise source to the output, both the amplitude of the noise and the gain for the noise sources at the significant harmonics needs to be calculated, and each noise source now has several contributions at the output from several different noise frequencies. The basic strategy is to apply the signals that cause nonlinearity in the hb analysis, and follow that with an hbnoise calculation.

This same approach can be taken for mixers where the LO signal deliberately introduces nonlinearity for the purpose of frequency translation. The LO signal is applied to the mixer in the hb analysis, and hbnoise follows that.

This works with the mixer by itself, or in combination with other circuits, like an LNA, mixer, and a baseband amplifier.

Many designs need to tolerate a large amplitude blocking signal. Apply the LO and the blocking signal (and not the RF signal) in the hb analysis and follow with hbnoise. The blocker power can be swept in the hb analysis and hbnoise run after each sweep value for the blocker power to measure the desensitization.

For large-signal noise problems, or for systems that are not periodic, transient noise is provided. For more information on this, refer to the *Spectre Classic Simulator, Spectre APS, Spectre X, and Spectre XPS*. This approach requires specifying a maximum noise frequency to be analyzed, which in RF systems is usually quite high. Two timepoints are created in the period of noisefmax. The transient needs to be run over many noise points in order to characterize the noise present in the circuit and needs post-processing to see the

© 1994-2020 All Rights Reserved.

results in the frequency domain. This usually requires much more time than using small-signal noise analyses.

Overview of Simulation Capabilities

The fundamental quantity calculated in hbnoise is the output noise of the circuit with a large signal applied and including the noise translations caused by the nonlinearity of the circuit. In order to run hbnoise, a harmonic balance (hb) analysis needs to be run first in order to calculate the nonlinearity of the circuit and the amplitude of the harmonics that are present to mix with. Hbnoise takes this information from the hb analysis and then for each noise source it calculates the transfer functions from all the frequencies specified in the *Choosing Analyses* form. Then it calculates the frequency response of the noise source itself. Next, it calculates the total noise power at the output for that source at all the noise frequencies that contribute at the output. At the end, it adds up all the noise power from all the sources at all the frequencies. Input-referred calculations require the selection of the input frequency range by setting the reference sideband. This selection is necessary because there are many transfer functions from the input to the output caused by mixing or aliasing from the different harmonics of the system. The passband frequency is needed for the noise figure calculations.

Note that hbnoise is a small-signal calculation that works for periodic systems. If the noise is large-signal, or the system is not periodic, transient noise should be used.

© 1994-2020 All Rights Reserved.

Example

Consider the double-balanced diode mixer below where a single-sideband noise figure measurement is desired.

The input source has variables set to define the frequency and amplitude of the two large-signal outputs. This is done so that the frequency and the amplitude can be set in the ADE Explorer window without changing the schematic.

© 1994-2020
All Rights Reserved.

The property list for the input port is shown below.

© 1994-2020
All Rights Reserved.

The variables have values set in the Design variables section. The easiest way to get the variables list is to generate a netlist from ADE Explorer or use *Variables - Copy From CellView*. This will cause all the variable names to be entered in the variables section. Just click in the value field for each variable and enter a value. Press *Enter* when you are done. In the figure below, the LO frequency set by the variable flo is 1GHz. The variable `vrf` which sets the linear amplitude of the large-signal RF inputs is zero. This disables the large-signal waveform. This circuit just has the signal that causes the frequency translations to occur (the LO signal) applied. Note that 1.4 volts peak is +13 dBm, which is a common drive level for a passive diode mixer.

The hb analysis needs to be run first in order to measure the amplitude of the harmonics and to characterize the nonlinearity of the system. From this, the noise folding from all the noise sources can be calculated in the hbnoise analysis. For more information on the setup of the hb form, please refer to the hb large-signal analysis section.

© 1994-2020 All Rights Reserved.

In the form below, the 1GHz LO is solved using automatic harmonics.

The noise output frequency and the number of noise translations to be calculated are set up in the hbnoise form. In the example below, the output frequency is near baseband. Leaving

© 1994-2020
All Rights Reserved.

the *Maximum sideband* field blank causes noise translations from all the harmonics in the hb analysis to be calculated.

The reference sideband selects the passband frequency for the noise figure calculation.

© 1994-2020
All Rights Reserved.

Once both forms are set up, the simulation is run. To plot outputs, select *Results - Direct Plot - Main Form* in the ADE Explorer window. The top of the form lists all the analyses that were run. Select *hbnoise* where the noise figure is calculated. Next, select *NF* because that is what is desired to plot. If the noise figure over a bandwidth is desired, select *Integrated Over Bandwidth*, and specify the frequency range you desire for the measurement.

When you are done, click *Plot*.

© 1994-2020
All Rights Reserved.

The waveform window plots the noise figure and the noise figure over the range you specified to be displayed, as shown below.

Noise Output Near Zero Frequency

First, the hb analysis is run to calculate the nonlinearity and the harmonics. Then hbnoise is run to calculate the noise. In all systems there is a maximum number of noise translations that need to be taken into account. Either the system harmonics become so small that the amount of noise translation becomes minimal or the devices themselves have finite noise bandwidths.

© 1994-2020
All Rights Reserved.

These noise calculations from mixing or aliasing with the different harmonics are called sidebands.

In the diagram above the X axis is frequency. The top represents the harmonics that are calculated by the hb analysis. The bottom represents the noise output frequency. The arrows show the various mixing paths to the output frequency. Hbnoise calculates the noise contribution at the output with all these mixing paths in place. Black is the noise that occurs without mixing. Purple is the noise that mixes with the first harmonic. Blue is the noise that mixes with the second harmonic. Red is the noise that mixes with the third harmonic.

In hbnoise, the output frequency (which is usually a range of frequencies) is specified along with the highest harmonic to calculate the mixing products.

© 1994-2020 All Rights Reserved.

Noise Output Near The First Harmonic

Here the noise output frequency is shifted up to just above the first harmonic frequency. Black is the noise that occurs without mixing. Purple is the noise that mixes with the first harmonic. Blue is the noise that mixes with the second harmonic. Note that the noise frequency just below the first harmonic mixes with the second harmonic to provide the output just above the first harmonic. Red is the noise that mixes with the third harmonic. Note that the second term that mixes with the third harmonic is just above the fourth harmonic of the hb analysis which is not shown. The noise term shown with a dotted line is mixing with the fourth harmonic of the system.

Frequency Sweep

Since the fundamental quantity that is calculated in hbnoise is the output noise, the frequency range defined in the *Choosing Analyses* form defines an output frequency range to calculate the noise for. For driven circuits, the default is to sweep an absolute frequency range. Oscillators will be covered later.

For an up-conversion system, sometimes a log frequency range is desired near one of the harmonics of the system. In that case, select relative sweep, and specify a harmonic number. In the output frequency range, type in the desired offset frequency from that harmonic. Note that this is only available for frequencies above the desired harmonic number because to get this below the harmonic requires taking the log of a negative number, which does not exist.

© 1994-2020 All Rights Reserved.

When automatic is selected for sweep type, 100 total points will be calculated. In most cases, fewer points are needed so it is usually better to select linear or log and then specify the sweep you desire. Typically 3 to 5 points per decade or about 20 total points are needed.

Maximum Sideband

There is a property in the *Choosing Analyses* form called *Maximum sideband* that is used to define how many mixing products should be calculated. If *Maximum sideband* is zero, the noise analysis will only contain the noise that appears without mixing or aliasing of any type. This is not equivalent to a linear noise analysis because hbnoise takes into account the instantaneously varying nature of the noise because of the large signal being present in the hb analysis.

When *Maximum sideband* is one, then the noise without frequency translation and the noise that mixes or aliases with the first harmonic is present in the noise analysis. When *Maximum sideband* is 10, the noise analysis includes noise mixing through the 10th harmonic of the harmonic balance analysis. Usually the number of harmonics in the hb analysis and the maximum number of sidebands should agree. This is easily accomplished by leaving the *Maximum sideband* field blank in the *Choosing Analyses* form in ADE Explorer.

Setting Harmonics and Sidebands

The process of setting harmonics and sidebands is similar to setting the maximum number of harmonics for the harmonic balance simulation. Start with an estimate based on the harmonic content of the input signal(s) and the nonlinearity of the circuit. Then run a simulation. Raise harmonics and sidebands and run again. If the noise measurement did not change, then the original number of harmonics and sidebands might be able to be reduced. If it did change, then more are needed. Getting a stable hbnoise output may take more or fewer harmonics and sidebands than the harmonic balance simulation by itself.

Noise Separation

In addition to the total output noise, the individual noise contributions can be plotted if noise separation is selected in the *Choosing Analyses* form. More information about the noise separation will be provided later. Noise separation is available when the noise type is set to *timeaverage*. The things that can be plotted are:

1. Total noise at the output from each individual input frequency.
2. The noise at the output from each instance name with all noise mechanisms included for mixing from an individual input frequency.

© 1994-2020 All Rights Reserved.

3. The noise at the output from each instance name with all noise mechanisms broken out separately for mixing from an individual input frequency.
4. The current noise at the noise instance for each instance name with all noise mechanisms included for mixing from an individual input frequency.
5. The noise current in the instance from each instance name with all noise mechanisms broken out separately for mixing from an individual input frequency.
6. The gain to the output from each individual instance with all the individual noise sources broken out separately for mixing from an individual input frequency.

Multiple hbnoise

Some systems contain multiple points in the circuit where the noise measurement is desired. When the *Noise Type* is *timeaverage*, multiple hbnoise analyses are allowed. At the top of the hbnoise *Choosing Analyses* form, an option called *Multiple hbnoise* is provided. When this option is selected, up to six hbnoise analyses can be made after a single hb analysis.

Hbnoise after Multi-Tone hb

When sampled(jitter) is selected, any number of sampled measurements can be made.

Note: Currently, you cannot have a mix of *timeaverage* and *sampled*, but this will be made available in a future release.When multiple input frequencies are present in the circuit, there are more frequencies in the large-signal analysis to mix with. Not only are the harmonics of both tones present, the intermodulation products are present as well. In this case, the noise analysis becomes more involved because the transfer functions from all the frequencies present in the large-signal have to be calculated for each noise source. Also, the meaning of *Maximum sideband* slightly changes. If 5 were selected, all the noise translations would be calculated through the 5th harmonic of the tone with tstab enabled. In the harmonic balance *Choosing Analyses* form it is *Tone1* when *Frequencies* is selected, and the tone with tstab set to *yes* in the *Names* list. It is recommended that the largest number of harmonics for any tone in the hb list be used as maximum sideband. This can easily be accomplished by leaving the number of sidebands field being left blank.

Note that noise type=timeaverage is the only available choice for a 2-tone driven circuit.

AM/PM Noise for Driven Circuits and Oscillators

Starting with IC617, the noise UI has been re-designed. *modulated* has been removed and its functionalities have been merged to *timeaverage*, leaving three noise types now. In

© 1994-2020 All Rights Reserved.

addition, the FM jitter option has been removed. It is calculated through timeaverage PM noise, which should be used when you want to get phase noise.

When *Noise Type* is set to *timeaverage*, USB, LSB, AM and PM noise are directly available.You can choose, for example, PM noise alone, or you can select the ALL(AM,PM,USB,LSB) option, if you need all four types of noise. The available options are *USB*, *AM*, *PM*, *AM&PM*, and *ALL(AM,PM,USB,LSB)*.

USB denotes the total noise in the frequencies above the fundamental frequency and LSB denotes the noise below the fundamental frequency.

For more information on changes, refer to the blog *Noise Simulation in Spectre RF Using Improved Pnoise/Hbnoise and Direct Plot Form Options*.

AM and PM noise are discussed below.

To illustrate AM noise, see the phasor diagram below.

This diagram illustrates a periodic signal with AM noise. The black phasor is the output signal, and AM noise is a random vector that varies the amplitude of the signal.

© 1994-2020
All Rights Reserved.

PM noise is shown in the phasor diagram below.

In this case, the amplitude is constant, but the noise is a random vector perpendicular to the signal phasor. The timing of the signal varies a bit from cycle to cycle.

The FM jitter and phase noise are only available through the PM noise option.

© 1994-2020 All Rights Reserved.

Most circuits produce a combination of AM and PM noise. This concept is shown below.

In this case, a random vector at any angle and amplitude is produced that varies at each instant of time. If the uncertainty creates a round shape, the AM component and the PM component are the same size.

© 1994-2020
All Rights Reserved.

In many circuits, AM or PM noise may dominate. If the PM noise dominates, the circle turns into an ellipse that is large in the perpendicular direction than in the parallel direction. This is shown below.

Jitter

Jitter is limited to single input frequency circuits. Jitter is usually applied to digital circuits where the average noise over the entire waveform is not appropriate. The noise in the high state and in the low state does not matter because the output voltage is so far from the voltage that triggers the load that it does not matter. The only time the noise matters is at the exact time where the trigger event is generated. Also, the noise in the low and high state is quite small because one device is completely off, and the other device is driven on. It is basically acting as a resistor. The device is not very good as an amplifier because the output is pulled hard into a rail. As the state changes, both devices are on, and they are relatively better amplifiers, so the noise is likely to be larger in the switching interval. In jitter for driven circuits, the voltage threshold for the noise measurement, and the selection of the rising edge, the falling edge, or both is made. At the threshold voltage specified, a noise measurement is made at that instant. Once the noise voltage is known, the timing uncertainty can be

© 1994-2020 All Rights Reserved.

determined from the slew rate. Since the RMS noise voltage is calculated, the timing uncertainty is one standard deviation from the ideal crossing time. This is illustrated below.

In most cases, shooting PSS will provide a faster simulation that is much less prone to error. If you must use harmonic balance, ensure that you have enough harmonics in the HB simulation in order to get the correct slope of the above waveform. If you under specify the number of harmonics, the slope will be smaller and the jitter measurement will be correspondingly larger. You can get an estimate of how many harmonics you need from the tstab waveform by taking the period of the signal divided by the fastest 10% to 90% rise or fall times in the circuit.

sampled(jitter) has three modes:

- Edge crossing mode: Allows a jitter measurement at a time specified by a threshold specification and rise or fall direction. The measurement can be made on that same net or at another net in the circuit. This is provided so the switch clock of a switched-capacitor circuit can specify the time for a noise measurement at the output of the filter.
- Sampled phase mode: This is like the old timedomain hbnoise. In this mode, you can specify a list of times in the hb timedomain waveform and the noise measurements will be made at the specified times.
- Edge delay mode: Allows both the trigger and the measurement to be edges anywhere in the circuit. You can use the rising edge of a pulse as the trigger and the falling edge of the same pulse to be the measurement, and then you will get the jitter on the pulse. You can also set the input to an inverter as the trigger and the output of the inverter as the measurement, in which case you will measure the jitter of the inverter.

© 1994-2020 All Rights Reserved.

Oscillators

timeaverage USB noise

For oscillators, the hbnoise frequency range defaults to *relative*. Specify the harmonic number as appropriate for the system you are simulating. If you are simulating an oscillator by itself, then the harmonic number is likely to be 1. If you have an oscillator and a diode frequency doubler, then the harmonic number is likely to be 2. If you have an oscillator with a frequency divider, in the hb form, you should specify the approximate frequency of oscillation for the oscillator itself, and set the freqdivide property to the divider ratio. This has the effect of dropping the first harmonic frequency by the divide ratio. In the hbnoise form, if the noise is desired on the frequency divided output, then the relative harmonic is 1. If the noise is desired at the output of the oscillator, the relative harmonic number is the divide ratio. The meaning of relative is to take the frequency of harmonic number specified and add to it the frequencies specified in the Choosing Analyses form. If the oscillator had a 1GHz output, and the hbnoise had 1M relative to the first harmonic specified, the actual output frequency is 1G + 1M, or 1001M.

Oscillators do not need noise figure calculations and are not typically driven. The output is usually a voltage, and an output node is typically selected.

Hbnoise is a small-signal analysis and is not limited by large-signal effects such as clipping or slew-rate limits. As a result, at low offset frequency, the phase noise might be significantly greater than 0dBC/Hz. This indicates that the noise is larger than the oscillations, which is not physically possible. If you want to see the phase noise curve level off at low frequency, set the *lorentzian* option to *yes*.

Note: The *lorentzian* option is supported only when *Noise Type* is set to *timeaverage*.

sampled(jitter)

This is like the jitter measurement described above.

Noise Calculations in the Simulator

Noise Factor and Noise Figure

When the hbnoise analysis is set up with voltage selected instead of probe for the input, noise figure and noise factor cannot be calculated because the system resistance specified in the

© 1994-2020 All Rights Reserved.

port component is not available to the simulator. Using a port as the input probe is required to obtain noise figure and noise factor calculations.

In the *Direct Plot Form*, noise factor and noise figure are both single-sideband measurements.

Noise factor is:

```
((All the noise power at the output from all the components with all the frequency
translations) - (Noise power at the output from the load))/Noise power at the output
from the resistance of the input port from the passband frequency only.
```

In the *Choosing Analyses* form, the input port is identified by the instance name. The port component is a voltage source in series with a resistor and is considered as a single element. No access is provided to the results for the internal node where the resistor and voltage source join.

In order to determine the passband frequency, the reference sideband must be supplied. This specifies the passband frequency for the noise figure calculation. The easiest way to select the reference sideband is to use the *Select from list* feature in the hbnoise *Choosing Analyses* form.

The noise from the load is normally defined by selecting the port or resistor that is serving as the load for the circuit in the Choosing Analyses form.

Noise figure is:

```
10*log(10) (Noise Factor)
```

Noise factor (IEEE) is similar to the noise figure measurement above. If you subtract from the numerator all the noise from the input port at all frequencies except the passband, you have the IEEE noise factor.

As mentioned earlier, in order to calculate the IEEE noise factor, a port component must be used as the input. In the *Choosing Analyses* form, the input port is identified by the instance name. The port component is a voltage source in series with a resistor and is considered as a single element. No access is provided to the results for the internal node where the resistor and voltage source join.

In order to determine the passband frequency, the reference sideband must be supplied. This specifies the passband frequency for the noise figure calculation. The easiest way to select the reference sideband is to use the *Select from list* feature in the *Choosing Analyses* form.

The noise from the load is normally defined by selecting the port or resistor that is serving as the load for the circuit in the *Choosing Analyses* form.

© 1994-2020 All Rights Reserved.

Noise Factor IEEE is:

```
((All the noise power at the output from all the components with all the frequency
translations) - (Noise power at the output from the resistor in the input port at
all frequencies except for the passband) - (Noise power at the output from the
load))/Noise power at the output from the resistance of the input port from the
passband frequency only.
```

Noise figure IEEE is:

```
10*log(10) (IEEE Noise Factor)
```

Noise factor DSB is similar to the single sideband noise factor above, except in the denominator, the noise power from the input port at the image frequency is added.

Noise factor (DSB) is:

```
((All the noise power at the output from all the components with all the frequency
translations) - (Noise power at the output from the load))/Noise power at the output
from the resistance of the input port from the passband and image frequencies only.
```

As mentioned earlier, in order to calculate the DSB noise factor, a port component must be used as the input. In the *Choosing Analyses* form, the input port is identified by the instance name. The port component is a voltage source in series with a resistor and is considered as a single element. No access is provided to the results for the internal node where the resistor and voltage source join.

In order to determine the passband frequency, the reference sideband must be supplied. This specifies the passband frequency and by inference the image frequency for the noise figure calculation. The easiest way to select the reference sideband is to use the *Select from list* feature in the *Choosing Analyses* form.

The noise from the load is normally defined by selecting the port or resistor that is serving as the load for the circuit in the *Choosing Analyses* form.

Noise figure (DSB) is:

```
10*log(10) (Noise Factor (DSB))
```

Summary:

`No` = total output noise
`Ns` = noise at the output due to the input probe at the passband frequency (the source)
`Nsi` = noise at the output due to the image harmonic at the source
`Nso` = noise at the output due to harmonics other than input at the source
`Nl` = noise at the output due to the output probe (the load)
`IRN` = input referred noise
`G` = gain of the circuit
`F` = noise factor
`NF` = noise figure

© 1994-2020 All Rights Reserved.

`Fdsb` = double sideband noise factor
`NFdsb` = double sideband noise figure
`Fieee` = IEEE single sideband noise factor
`NFieee` = IEEE single sideband noise figure

then,

```
IRN = sqrt(No^2/G^2)
F = (No^2 - Nl^2)/Ns^2
NF = 10*log10(F)
Fdsb = (No^2 - Nl^2)/(Ns^2+Nsi^2)
NFdsb = 10*log10(Fdsb)
Fieee = (No^2 - Nl^2 - Nso^2)/Ns^2
NFieee = 10*log10(Fieee)
```

When the hbnoise analysis is set up with voltage selected instead of probe, noise figure cannot be calculated.

Input-Referred Noise

When the hbnoise analysis is set up with the input source being a port, input-referred noise may not be exact. In this case, the input-referred noise is referred to the voltage source that is inside the port component. The amplitude of the noise is divided by two, which assumes a perfect match. If an accurate input-referred noise is desired, set the *Input Source* to *voltage*, and select the input node. This will disable the noise figure calculation.

Noise transfer function

Hbnoise calculates the forward gain from the passband to the output and saves it. In the *Direct Plot Form*, this is called *Transfer Function*, and is in linear volts at the output divided by volts at the input.

© 1994-2020 All Rights Reserved.

Commonly Used Options

Tolerance

Leave this option at the default value.

© 1994-2020
All Rights Reserved.

Hbnoise uses an iterative solver to calculate the output amplitudes. Any iterative solver needs an error tolerance to specify when to stop iterating because the solution is accurate enough, and the tolerance option specifies that accuracy for hbnoise. For driven circuits, the default is 1e-6. For oscillators, the default is 1e-4.

Resgmrescycle

Leave this option at the default.

Hbprecond_solver

The basic solver is the only solver available in standard Spectre. Autoset is the default solver when APS is used. This solver is faster. If stagnation is detected, Spectre prints a message in the Spectre output window.

Annotate

This option controls the amount of output in the spectre output window. Selecting *title* produces the least amount of detail, and selecting *steps* gives the most amount of detail. The options between produce more details as you move from left to right.

Stimuli

This option is not implemented in the *Direct Plot Form* in ADE Explorer. It is used for the hbxf analysis that runs when an hbnoise is run. The default is *sources* which calculates the forward conversion gain in hbxf from the sources in the circuit only. *nodes_and_terminals* also calculates the conversion gain from all the nodes in the circuit to the output. To display these results in ADE Explorer, you must use the results browser and the calculator.

Save

The default is *allpub*. This saves all the public voltages at all levels of the hierarchy in the schematic. It excludes the internal nodes of the device models. *all* adds the internal nodes of all the devices. *lvl* saves all the nodes including the internal nodes of the devices through the level of hierarchy set in the *nestlvl* option. *lvlpub* is like *lvl*, but it does not save the internal nodes. *selected* saves only the nodes that are specifically saved. In ADE Explorer, this is accomplished using *Outputs - To Be Saved - Select On Design* and then selecting the nodes and terminals specifically in the schematic. At the netlist level, this is accomplished by using a save statement with a list of names to be saved.

© 1994-2020 All Rights Reserved.

Nestlvl

If *save* is set to *lvl* or *lvlpub*, this controls the maximum level of hierarchy to be saved. If *nestlvl* is *1*, only the top level is saved. If *nestlvl* is 2, the top level and the next level down are saved. The value for *nestlvl* can be any integer. The example below shows *lvlpub* selected that saves two levels of hierarchy.

Saveallsidebands

Leave this option at the default value of *no*. If you want to see the noise contributors from the different noise input frequencies, select *yes* for noise separation in the hbnoise *Choosing Analyses* form.

Lorentzian

This option is documented in the oscillator section. It only affects oscillators.

AdditionalParams

This is a field for `<option_name>=value` statements. Multiple `<option_name>=value` statements are allowed with a space between them. Generally, this is used to unlock the beta features.

ADE Explorer Implementation

This section jumps from theory to practical with examples on how to use the individual analysis types that have been covered. Driven circuits and oscillators will be shown.

Driven Circuits

Noise Type = timeaverage USB noise

On the *Choosing Analyses* form:

© 1994-2020 All Rights Reserved.

1. First, set up the harmonic balance analysis. For more information, see the Harmonic Balance section at the beginning of this chapter.

© 1994-2020
All Rights Reserved.

2. In the hbnoise *Choosing Analyses* form, specify the output frequency range of your circuit.

Note: Since the fundamental quantity that is calculated by the noise analysis is the noise at the output, the frequency range is always the output frequency range.

3. Set about 3 to 5 points per decade in the *Points Per Decade* field.

4. Leave the *Maximum sideband* field blank or set it to the same number as the number of harmonics in the hb analysis. This specifies that the noise that mixes with all the hb harmonics is calculated.

5. Specify the output of the circuit. When the output is set to *probe*, a resistor or a port can be selected and this noise will be subtracted from the noise figure calculation.

6. Select *port* from the *Input Source* drop-down list for noise figure calculation.

© 1994-2020
All Rights Reserved.

7. Choose *Select from list* from the drop-down list for reference sideband, which specifies the input frequency range for the noise figure calculation.

8. Select the passband frequency range from the list.

9. Select *timeaverage* from the *Noise Type* drop-down list to calculate the average noise power at the output averaged over the entire time of one cycle of the input waveform.

10. Select the check box for *Noise Separation* to enable plotting of the individual noise contributors in the circuit. Noise separation is only available when *Noise Type* is set to *timeaverage.*

Multiple hbnoise

When *Noise type* is set to *timeaverage*, multiple hbnoise analyses can be run after a single hb analysis. This is useful for LNA-Mixer combinations, or in any circuit where there are multiple points in the circuit where noise is important.

To run multiple analyses after a single hb:

© 1994-2020 All Rights Reserved.

1. First, set up the harmonic balance simulation. for more information, see the harmonic balance section at the beginning of this chapter.

© 1994-2020
All Rights Reserved.

2. Select the *Multiple hbnoise* check box located at the top of the hbnoise *Choosing Analyses* form.

3. Set up all the noise analyses you want to run, as shown below.

© 1994-2020
All Rights Reserved.

Noise Type = timeaverage AM, PM, AM&PM

timeaverage also allows the measurement of the AM and PM components of noise separately.

First, set up the hb analysis as appropriate for your circuit. For more information, see the harmonic balance section of this chapter.

On the hbnoise *Choosing Analyses* form:

© 1994-2020
All Rights Reserved.

1. Set up an appropriate frequency range of interest.

2. Select *timeaverage* from the *Noise Type* drop-down list.

3. Select the ALL(AM,PM,USB,LSB) radio button.

   When you select the ALL(AM,PM,USB,LSB) option, Sweeptype automatically changes to relative.

4. Specify a number in the *Relative Harmonic* field.
5. Run the simulation.

Now plot the Output noise using the *Direct Plot Form*.

On the *Direct Plot Form*:

1. Select *Output Noise* from the *Function* section.

© 1994-2020 All Rights Reserved.

2. Select *AM* or *PM* from the *Noise Type* radio buttons.
3. Enter the frequency in the *Value At (Hz)* field if you want to plot the value at a specific frequency. You can enter a list of frequencies separate by a space in this field.
4. Choose SSB or DSB for AM and PM noise. DSB is 3dB higher than SSB noise.

© 1994-2020 All Rights Reserved.

Note the DSB AM noise is 3dB larger than the output noise. This is because the output noise is derived from the positive frequencies only, and the DSB AM and PM noise is derived using the double sidebands.

5. If you want this signal to plot automatically when a new simulation is run, select *Add To Outputs*. The expression is automatically evaluated in ADE.

Noise Type = sampled(jitter)

Driven jitter is usually used for measuring the noise in a digital circuit at a specific threshold crossing. In this mode you must use enough harmonics to get a correct slope of the waveform. This must be done manually. Auto harmonics might not give you enough harmonics.

On the hb Choosing Analyses form:

© 1994-2020 All Rights Reserved.

1. Set a variable for the *Number of Harmonics* field.

© 1994-2020
All Rights Reserved.

2. Set the variable to a small number in ADE and run the simulation.

3. Open *Results - Direct Plot - Main Form* and plot the timedomain waveform with *Add to Outputs* selected in the Direct Plot Form.

4. Run a sweep in ADE.

   After the simulation is complete, a family of waveforms will be plotted. Zoom in to the rising and falling edges to view the waveform details.

© 1994-2020
All Rights Reserved.

© 1994-2020
All Rights Reserved.

The rising edge needs more harmonics than the falling edge. Approximately 40 harmonics are needed in the example shown above.

Disable the sweep and set the variable to the number of harmonics you determine. In this example, it was set to 40.

On the hbnoise *Choosing Analyses* form, the stop frequency is half the input frequency. The start frequency is four decades below that. About five points per decade is usually sufficient. *Maximum sideband* should be left blank.

Edge Crossing Mode

In this mode, you specify a trigger which is an edge in the circuit. This determines the time in the waveform to make the measurement. The measurement is specified separately in order to allow one waveform to set the time and measure another net at that time.

Click *Add*. The threshold is set appropriately. The *Edge Direction* option can be set to *rise* or *fall*.

© 1994-2020 All Rights Reserved.

You can also specify *Sleep Time*, which is the length of time that Spectre will wait before checking for the crossing event in pnoise and hbnoise sampled(jitter) analysis. It is netlisted in Spectre as "delay". This is used in circuits (such as dividers) where there are multiple edges in the PSS and hb periodic steady state interval.

© 1994-2020
All Rights Reserved.

When the simulation completes, use the *Direct Plot Form* to plot the jitter. For driven circuits, select *Jee*, which is edge-to-edge jitter. In the below example, 77p is the rising edge, and 275p is the falling edge.

© 1994-2020
All Rights Reserved.

The output noise frequency response and the edge-to-edge jitter are displayed. The jee measurement is one standard deviation for the expected jitter.

Sampled phase mode

First, plot the timedomain waveform for the net that you want to measure. Place markers at the lowest and highest voltage that you want to explore.

© 1994-2020
All Rights Reserved.

In this case, times from 72psec to 84psec would capture the variation in noise voltage from 1 Volt below the threshold to 1 Volt above the threshold.

On the hbnoise *Choosing Analyses* form:

© 1994-2020 All Rights Reserved.

1. In the hbnoise *Choosing Analyses* form, select Sampled Phase

© 1994-2020
All Rights Reserved.

2. Set the stop frequency to half the input hb frequency.

3. Set the start frequency to four decades smaller than the stop frequency.
4. Leave the *Maximum sideband* field blank.
5. Select the net to be measured.
6. Type *1* (one) in the *Samples Per Period* field.
7. Select the *Add Specific Points* check box.
8. Type a list of times in the *Add Specific Points* field.
9. Click *Add*.
10. Click *OK*.

© 1994-2020
All Rights Reserved.

11. Run the simulation and open the Direct Plot Form.

12. If required, select the appropriate noise analysis in the *Measurement Selection* drop-down list.

13. Select *HB Integ Output Noise*.

14. Set the frequency range to the values used in hbnoise.

15. If required, click *Add To Outputs*.

16. Click *Plot*.

© 1994-2020 All Rights Reserved.

The noise waveform appears in another subwindow.

17. To put the noise waveform in the same subwindow as the output waveform, right-click the X Axis in the left subwindow.

18. Select *Allow Any Units*.

19. Drag the noise waveform into the same subwindow as the waveform and delete the other subwindow.

© 1994-2020
All Rights Reserved.

20. Zoom in to the area of interest.

The rising edge and the noise voltage are plotted.

© 1994-2020
All Rights Reserved.

Edge Delay Mode

1. In the hbnoise Choosing Analyses form, set the stop frequency to half the PSS frequency.

2. Set the start frequency to four decades below that.

© 1994-2020
All Rights Reserved.

3. Select *Edge Delay* for *Timing Event*.
4. Select the net that starts the jitter measurement time.
5. Set the appropriate voltage for *Threshold Value*.
6. Select *Rise* or *Fall* from the *Edge Direction* drop-down list.
7. (Optional) Set *Sleep Time*, which is the length of time that Spectre will wait before checking for the crossing event in hbnoise sampled(jitter) analysis.
8. Select the net that stops the jitter measurement.
9. Select *Rise* or *Fall* from the *Edge Direction* drop-down list.
10. Click *Apply*.
11. Click *OK* to close the form.
12. Run the simulation.

© 1994-2020 All Rights Reserved.

13. After the simulation completes, select *Results - Direct Plot - Main Form*..

14. If you ran multiple analyses, select the appropriate hbnoise analysis in the *Measurement Selection* drop-down list.

15. Select the *Jdelay* from the Function section.

16. Set *Integration Limits* as the range you had set in the hbnoise analysis.

17. If required, select *Add To Outputs*.

© 1994-2020 All Rights Reserved.

18. Click Plot.

The jitter is shown in the marker attached to the trace.

Oscillators

Noise Type = timeaverage PM noise

The settings are the same as driven *Noise Type = timeaverage* with the exception of being a relative sweep by default and usually for oscillators the PM component of noise is desired. Since the phase noise of the oscillator is near one of the harmonics, *Sweeptype* is set to *relative* by default. A harmonic number is provided for the case where there is a frequency multiplier or a frequency divider on the output of the oscillator. *Noise type = timeaverage* also has the ability to plot the noise contributors from mixing with each harmonic when noise separation is selected. There is an example of noise separation later in this chapter.

The hbnoise setup is similar to driven except that the relative harmonic is specified and *PM noise* or *All* would be selected.

© 1994-2020 All Rights Reserved.

First, set up the harmonic balance analysis. For more information, see the Harmonic Balance section at the beginning of this chapter.

© 1994-2020
All Rights Reserved.

On the hbnoise *Choosing Analyses* form:

1. Select *relative* from the *Sweeptype* drop-down list.

2. Specify the appropriate harmonic number for your design. In this example, the first harmonic is specified in the *Relative Harmonic* field.
3. The frequency range should be the relative frequency range.

4. Leave the *Maximum sideband* field blank.

5. Select *voltage* from the *Output* drop-down list.

6. Specify the output node in the *Positive Output Node* field.

© 1994-2020
All Rights Reserved.

7. Select *timeaverage* from the *Noise Type* drop-down list. This gives an averaged noise measurement.

8. Select *PM*, *AM&PM*, or *ALL* to calculate the PM component of noise.
9. Select *yes* for *Noise Separation* if you want to plot the individual noise contributors.
10. Run the simulation.

© 1994-2020 All Rights Reserved.

11. After the simulation completes, select *Results - Direct Plot - Main Form*.

12. In the *Direct Plot Form*, select *hbnoise* from the *Analysis* section.

13. Select *Output Noise* from the *Function* section.

14. Select *dBc/Hz* from the *Units* section.

© 1994-2020 All Rights Reserved.

15. (Optional)To measure phase noise at different offset frequencies using an expression, type those frequencies in the *Value At (Hz)* field.

Separate the entries with a space. When you also select the *Add To Outputs* check box, the expressions will be entered in ADE that measure the phase noise at all the frequencies in the list.

16. Select *Noise convention* as needed.

17. Click *Plot*.

The phase noise is plotted.

© 1994-2020
All Rights Reserved.

18. To plot the cycle or cycle-to-cycle jitter, select *Jc* or *Jcc* from the *Function* section.

19. Specify the start and stop frequencies as shown in the figure above.

20. *Click* Add To Outputs.

© 1994-2020
All Rights Reserved.

An expression to measure the cycle jitter result is entered into ADE.

NoiseType=sampled(jitter)

If you have square wave circuits, pss shooting is likely to run faster and be less error prone than harmonic balance. For more information, see the discussion in the driven circuit sampled(jitter) section.

This capability is seldom used for oscillators. There is the ability to plot the single-sideband phase noise from the Direct Plot Form, however the result will usually agree with timeaverage except at high frequency offsets. When using hbnoise, you must take care to have enough harmonics to accurately capture the output waveform. Refer to the sampled(jitter) for driven circuits section to know more about the process.

All the modes available for driven circuits are available for oscillators. The only mode with value for oscillators is edge crossing mode. The other modes will not be covered in this section. To know more about the other modes, refer to the sampled(jitter) driven circuits section. First, set up the harmonic balance analysis. Ensure that you have enough harmonics

© 1994-2020 All Rights Reserved.

to accurately capture the slope of the output waveform. For more information, see the harmonic balance section at the beginning of this chapter.

© 1994-2020
All Rights Reserved.

On the hbnoise *Choosing Analyses* form:

1. Select *sampled(jitter)* from the *Noise Type* drop-down list.

2. Select the output net.
3. In the *Threshold Value* field, specify the voltage for instantaneous noise measurement. If you are not sure of the appropriate voltage, run the hb analysis, and plot the timedomain waveform for the output node(s).
4. (Optional) Specify *Sleep Time*, which is the length of time that Spectre will wait before checking for the crossing event in hbnoise sampled(jitter) analysis.
5. Select a value from the *Edge Direction* drop-down list.
6. Select *Enabled*.
7. Run the simulation.

© 1994-2020 All Rights Reserved.

8. After the simulation completes, select *Results - Direct Plot - Main Form.*

9. Select *hbnoise sampled* from the *Analysis* section.
10. If you ran multiple analyses, select the appropriate run from the *Measurement Selection* drop-down list.
11. Select *Edge Phase Noise* for the *Function* section.
12. Click *Plot*.

© 1994-2020
All Rights Reserved.

The waveform appears. Note that this is single sideband phase noise.

© 1994-2020
All Rights Reserved.

Examples

Driven Circuit Noise Setup

First, set up the harmonic balance analysis. For more information, see the Harmonic Balance section at the beginning of this chapter.

© 1994-2020 All Rights Reserved.

© 1994-2020
All Rights Reserved.

On the hbnoise *Choosing Analyses* form:

1. Specify the output frequency range of your circuit.

2. If you use log spacing, set about 3 to 5 points per decade in the *Points Per Decade* field.

3. In hbnoise, and pnoise or qpnoise when Harmonic Balance is set for the pss or qpss engine, you can leave the *Maximum sideband* field blank. Then, all the mixing products from the large-signal analysis will be considered.

4. If you want a noise figure calculation, select the *Noise Figure* check box.

© 1994-2020
All Rights Reserved.

5. Select the resistor or port component used for the load resistance. This excludes the noise from this component for the noise figure calculation, but keeps it for the total noise output calculation.

6. If you want a noise figure calculation, you must use a port as the input for the circuit. Set the instance name.

7. The easiest way to select the reference sideband is to set it to choose *Select from list*.

8. Specify the passband frequency range that you want for the noise figure and input-referred noise calculations.

© 1994-2020
All Rights Reserved.

9. If you want to see the noise contributions from the individual frequencies and the different device noise calculations individually on the output of the noise analysis, enable noise separation.

10. When you are done with the setup, run the simulation.

Noise Summary

The noise summary provides information about the devices that contribute noise to the output. It supports all USB, LSB, AM and PM noise. This information is available any time an hbnoise simulation completes.

In ADE Explorer, select *Results - Print - Noise Summary*. You will see the following options depending on the selection you have made for *Noise Type* in the hbnoise Choosing Analysis form

hbnoise_am - This option is available if you select the *timeaverage* option from the *NoiseType* drop-down list and select *AM* or *ALL(AM.PM.USB,LSB)* as the contribution type. *hbnoise_am* has information about the amplitude modulated noise.

hbnoise_pm - This option is available if you select the *timeaverage* option from the *NoiseType* drop-down list and select *PM* or *ALL(AM.PM.USB,LSB)* as the contribution type. *hbnoise_pm* as information about the phase modulated noise.

hbnoise_usb - This option is available if you select the *timeaverage* option from the *NoiseType* drop-down list and select *USB* or *ALL(AM.PM.USB,LSB)* as the contribution type. *hbnoise_usb* has information about the total noise from frequencies above the output frequency.

hbnoise_lsb - This option is available if you select the *timeaverage* option from the *NoiseType* drop-down list and select *ALL(AM.PM.USB,LSB)* as the contribution type.

© 1994-2020 All Rights Reserved.

hbnoise_lsb has information about the total noise from frequencies above the output frequency.

If noise separation is enabled, an additional *hbnoise_src* option will be added along with the above options. *hbnoise_src* has information about the noise currents at the individual noise sources at the noise source.

If you want the noise contributors at a single frequency, select *spot noise*. If you want to have the noise integrated over a frequency range, select *integrated noise*.

If you want noise in volts, set the *noise unit* to *V.*

You can choose a value from the *hierarchy level* drop down box to view the noise contributions from all noise sources within each instance at the specified level of hierarchy.

Generally, *Include All Types* should be selected. If you just want the noise from specific types of noise generators, you can select them from the list.

Truncate just applies to the list. The total input and output-referred noise at the bottom of the noise summary output always includes all the noise from everything in the circuit.

© 1994-2020 All Rights Reserved.

The figure below is the noise summary window with the sort being noise contributors. Then there are three windows that show the three different sorted outputs.

© 1994-2020
All Rights Reserved.

The output below shows the noise summary sorted by noise contributors. The largest noise contributor is on the top of the list.

Results Display Window

Window Expressions Info Help cadence

Device	Param	Noise Contribution	% Of Total
/Q6	rb	1.84571e-05	11.03
/Q7	rb	1.8327e-05	10.88
/Q2	rb	1.81054e-05	10.62
/Q3	rb	1.79802e-05	10.47
/R6	rn	1.66084e-05	8.93
/R5	rn	1.62943e-05	8.60
/R4	rn	1.62929e-05	8.60
/R1	rn	1.6292e-05	8.60
/Q2	ib	1.00577e-05	3.28
/Q6	ib	9.06092e-06	2.66
/Q2	fn	7.4014e-06	1.77
/Q3	ib	6.90555e-06	1.54
/Q6	fn	6.66657e-06	1.44
/Q2	ic	6.21759e-06	1.25
/Q6	ic	6.1849e-06	1.24
/Q7	ib	5.81623e-06	1.10
/Q3	ic	5.72646e-06	1.06
/Q7	ic	5.67656e-06	1.04
/Q3	fn	5.19519e-06	0.87
/Q7	re	4.40025e-06	0.63
/Q6	re	4.40019e-06	0.63
/Q7	fn	4.37609e-06	0.62
/Q3	re	4.3168e-06	0.60
/Q2	re	4.31674e-06	0.60
/R2	rn	3.93784e-06	0.50
/R3	rn	3.9353e-06	0.50
/rif	rn	3.23391e-06	0.34
/Q5	ib	2.76747e-06	0.25
/Q5	rb	1.4732e-06	0.07
/Q1	rb	1.47231e-06	0.07
/Q5	ic	1.27644e-06	0.05
/R9	rn	1.03033e-06	0.03

10

© 1994-2020 All Rights Reserved.

The output below shows the noise summary sorted by composite noise. The largest total noise instance is on the top of the list, and the individual noise contributors for that component are shown below the instance name.

Results Display Window

Window Expressions Info Help cadence

Device	% Of Total	Inp Ref Noise Param		Noise Contribution
/Q2	17.52	0.0137304	total	2.32602e-05
			rb	1.81054e-05
			ib	1.00577e-05
			fn	7.4014e-06
			ic	6.21759e-06
			re	4.31674e-06
			rc	1.62743e-08
/Q6	17.00	0.0135299	total	2.2909e-05
			rb	1.84571e-05
			ib	9.06092e-06
			fn	6.66657e-06
			ic	6.1849e-06
			re	4.40019e-06
			rc	1.61927e-08
/Q3	14.56	0.0125292	total	2.11988e-05
			rb	1.79802e-05
			ib	6.90555e-06
			ic	5.72646e-06
			fn	5.19519e-06
			re	4.3168e-06
			rc	1.54184e-08
/Q7	14.27	0.01241	total	2.09867e-05
			rb	1.8327e-05
			ib	5.81623e-06
			ic	5.67656e-06
			re	4.40025e-06
			fn	4.37609e-06
			rc	1.52868e-08
/R6	8.93	0.00983386	rn	1.66084e-05
			total	1.66084e-05
/R5	8.60	0.00964788	rn	1.62943e-05
			total	1.62943e-05
/R4	8.60	0.00964702	rn	1.62929e-05
			total	1.62929e-05
/R1	8.60	0.00964653	rn	1.6292e-05
			total	1.6292e-05
/R2	0.50	0.0023316	rn	3.93784e-06
			total	3.93784e-06

14

© 1994-2020 All Rights Reserved.

The output below shows the noise summary sorted by device name. The devices are listed in alphabetical order.

Device	% Of Total	Inp Ref Noise Param		Noise Contribution
/L0	0.00	0	total	0
			fn	0
			rn	0
/Q1	0.10	0.00102571	total	1.73233e-06
			re	0
			ic	9.10568e-07
			rc	5.44402e-11
			ib	6.42863e-08
			rb	1.47231e-06
/Q2	17.52	0.0137304	total	2.32602e-05
			re	4.31674e-06
			ic	6.21759e-06
			rc	1.62743e-08
			fn	7.4014e-06
			ib	1.00577e-05
			rb	1.81054e-05
/Q3	14.56	0.0125292	total	2.11988e-05
			re	4.3168e-06
			ic	5.72646e-06
			rc	1.54184e-08
			fn	5.19519e-06
			ib	6.90555e-06
			rb	1.79802e-05
/Q4	0.00	8.03558e-05	total	1.36208e-07
			re	2.35549e-08
			ic	4.34448e-08
			rc	3.39835e-10
			fn	4.73645e-08
			ib	7.7397e-08
			rb	8.87496e-08
/Q5	0.37	0.00200429	total	3.38505e-06
			re	0

For all of the noise summaries, here is a table of the abbreviations used in the *Param* column:

© 1994-2020 All Rights Reserved.

Resistor and Inductor:

Parameter	Meaning
`rn`	resistor thermal noise
`fn`	Flicker noise

Diode:

Parameter	Meaning
`id`	Diode current shot noise
`rs`	Series parasitic resistor noise

Gummel-Poon BJT:

Parameter	Meaning
`fn`	Flicker noise
`ic`	Collector current shot noise
`ib`	Base current shot noise
`rc`	Thermal noise from the collector parasitic resistor
`rb`	Thermal noise from the base parasitic resistor
`re`	Thermal noise from the emitter parasitic resistor

HiCUM:

Parameter	Meaning
`ib`	Base current shot noise

© 1994-2020 All Rights Reserved.

Parameter	Meaning
`ic`	Collector current shot noise
`rbi`	Thermal noise from the internal base parasitic resistor (See diagram below)
`rbx`	Thermal noise from the external base parasitic resistor (See diagram below)
`rcx`	Collector parasitic resistor thermal noise (See diagram below)
`re`	Emitter parasitic resistor thermal noise (See diagram below)
`fn`	Flicker noise

Figure 3-1 HiCUM Schematic Diagram

© 1994-2020
All Rights Reserved.

VBIC:

Parameter	Meaning
`rbi`	Thermal noise in resistor Rbi (See diagram below)
`rbx`	Thermal noise in resistor Rbx (See diagram below)
`rci`	Thermal noise in resistor Rci (See diagram below)
`re`	Thermal noise in resistor Re (See diagram below)
`rs`	Thermal noise in resistor Rs (See diagram below)
`itzf`	Collector current shot noise
`ibe`	Total base current shot noise
`ibex`	Node Bx to E base current shot noise (See diagram below)
`fn`	Ibe flicker noise (See diagram below)
`fnx`	Ibex flicker noise (See diagram below)
`iccp`	Parasitic transport collector to emitter current noise (See diagram below)
`ibep`	Parasitic transport base to emitter current noise (See diagram below)
`fnp`	Ibep flicker Noise (See diagram below)

© 1994-2020 All Rights Reserved.

Figure 3-2 VBIC Schematic Diagram

BSIM3:

Parameter	Meaning
`fn`	Flicker Noise

© 1994-2020
All Rights Reserved.

Parameter	Meaning
id	Thermal noise from the resistive drain current
rs	Thermal noise from the source parasitic resistor
rd	Thermal noise from the drain parasitic resistor

BSIM4:

Parameter	Meaning
fn	Flicker noise
id	Thermal noise from the resistive drain current noise
rs	Source parasitic resistor thermal noise
rd	Drain parasitic resistor thermal noise
igd	Gate-to-drain tunneling current noise
igs	Gate-to-source tunneling current noise
igb	Gate-to-bulk tunneling current noise
rbdb	Resistor thermal noise between dbNode and bNode (See diagram below)
rbpb	Resistor thermal noise between bNode and bNodePrime (See diagram below)
rbpd	Resistor thermal noise between Bulk and dbNode (See diagram below)
rbps	Resistor thermal noise between bulk and bNode prime
rbsb	Resistor thermal noise between sbNode and bNode (See diagram below)
rgbi	Gate Bias-Independent Resistor thermal noise

© 1994-2020 All Rights Reserved.

Figure 3-3 BSIM4 Schematic Diagrams

PSP-103

Parameter	Meaning
`sfl`	Channel current flicker noise
`sth`	Channel current thermal noise
`sig`	induced gate noise at 1 Hz

© 1994-2020 All Rights Reserved.

Parameter	Meaning
`sigth`	Induced gate noise
`shotgd`	Gate current shot noise from gate to drain
`shotgs`	Gate current shot noise from gate to source
`sjnoise`	Source to Bulk leakage current noise
`djnoise`	Drain to Bulk leakage and avalanche current noise
`rgatenoise`	Gate resistance thermal noise
`rbulknoise`	Bulk resistor thermal noise between node BP and BI (See diagram below)
`rjnoise`	Source side bulk resistor thermal noise between node BI and BS (See diagram below)
`rjundnoise`	Drain side bulk resistor thermal noise between node BI and BD (See diagram below)
`rwellnoise`	Well resistor thermal noise between node BI and B (See diagram below)
`rsourcenoise`	Source parasitic resistor thermal noise
`rdrainnoise`	Drain parasitic resistor thermal noise

© 1994-2020 All Rights Reserved.

Figure 3-4 PSP-103 Schematic Diagram

EKV 3.0

Parameter	Meaning
`fn_id`	Channel current flicker noise
`fn_ig`	Gate current flicker noise
`rs`	Source parasitic resistor thermal noise
`rd`	Drain parasitic resistor thermal noise
`rg`	Gate parasitic resistor thermal noise
`rsb`	Thermal noise for the substrate resistor connected between BSi and B. (See diagram below)
`rdb`	Thermal noise for the substrate resistor connected between BDi and B. (See diagram below)
`rbbs`	Thermal noise for the substrate resistor connected between BSi and Bi. (See diagram below where it is labeled R_{DSB}/2)

© 1994-2020 All Rights Reserved.

Parameter	Meaning
`rbbd`	Thermal noise for the substrate resistor connected between BDi and Bi. (See diagram below where it is labeled R_{DSB}/2)
`id`	Thermal noise of the channel resistor
`cor`	Induced gate noise
`shot_ig`	Gate current shot noise

© 1994-2020 All Rights Reserved.

Figure 3-5 EKV 3.0 Schematic Diagrams

For other model types:

1. In a UNIX/LINUX shell window type `which spectre`.

   This should return a path that ends in `/bin/spectre`.

© 1994-2020 All Rights Reserved.

2. Triple-click the line with the path, and then click the center mouse button to enter it as a command.
3. Use the backspace key, and delete the word `spectre`.
4. Type `cdnshelp` at the end of the expression, and press `Enter`.

   A browser window will be displayed.
5. Close the *Tip of the day* window.
6. At the top, you should see the release number of the simulator you are using.
7. Click the plus sign (+) to the left of the *MMSim* folder.
8. Click the plus sign (+) to the left of the *Spectre Circuit Simulator Components and Device Models Reference* folder.
9. Select the device type you are using.
10. At the bottom of that device listing, you will see *Component Statements*. Double-click on this.
11. Enter the name of the parameter in the search box in the right pane of the cdnshelp window.
12. Click the down arrow to the right of the *Find* field until you see the parameter name in the pane on the right side, and then see the definition of that parameter.

© 1994-2020 All Rights Reserved.

Plotting Noise Figure

On the *Direct Plot Form*:

1. Select hbNoise from the *Analysis* section.
2. Select the desired type of noise plot from the *Function* section.
3. If you want the noise figure over a specified bandwidth, check the *Integrated Over Bandwidth* checkbox.
4. Specify the frequency range you want for the noise figure calculation.

© 1994-2020 All Rights Reserved.

5. Click *Plot*.

The dsb noise figure and the noise figure calculation over your bandwidth are displayed.

Viewing noise separation results

Viewing the total noise at the output from a single noise input frequency

This capability allows the noise from all sources in the circuit at different noise input frequencies to be measured. For example, all the noise that does not change frequency (the zero sideband) can be plotted, along with the other noise input frequencies (sidebands). This plotting capability allows problem frequencies to be identified.

© 1994-2020 All Rights Reserved.

On the *Direct Plot Form*:

© 1994-2020
All Rights Reserved.

1. Select *hbnoise separation* from the *Analysis* section.
2. Select the noise type from the *Noise Type* section. By default, *USB* is selected.
3. To view the total noise at the output select *Sideband Output* from the *Function* section.
4. Select *V/sqrt(Hz)* or *v**2/Hz* from the *Signal Level* radio buttons.
5. Select the appropriate modifier from the *Modifier* section.
6. Select the sideband number form the *Output Sideband* list. 0 means no frequency translation.
7. Click *Plot*.

The output noise from sideband zero is plotted. This is the noise that is present at the output with no frequency translations included.

Plotting Noise from Sideband -1

On the *Direct Plot Form*:

© 1994-2020 All Rights Reserved.

1. Select the -1 sideband from the *Output Sideband* list. -1 means the noise frequency is shifted down in frequency by -1*first harmonic of hb from the frequency that is specified as the noise output frequency in the *Choosing Analyses* form.

© 1994-2020
All Rights Reserved.

2. Click *Plot*.

© 1994-2020
All Rights Reserved.

As a suggestion, use the *Graph - Add Label* menu option to place labels in order to identify the traces.

© 1994-2020
All Rights Reserved.

Tip

If you select multiple sidebands to be plotted at the same time, the sideband number will be shown at the right side of the legend in the waveform window.

© 1994-2020
All Rights Reserved.

Viewing total noise at the output from each instance

Once the problem frequencies are identified in the plot of *Sideband Output*, the problem components need to be identified. From the previous plot, the zero sideband contributes the largest amount of noise to the overall output noise. Using the *Instance Output* option allows the identification of which components in the zero sideband contribute the most noise. The total noise for all the noise mechanisms within the component is plotted.

© 1994-2020
All Rights Reserved.

On the *Direct Plot Form*:

1. Select *Instance Output* from the *Function* section.

© 1994-2020
All Rights Reserved.

2. Select *V/sqrt(Hz)* or *v**2/Hz*.
3. Select the appropriate modifier.
4. Select the sideband number.
5. Select the type of component to display.
6. If you want every noise source to be considered, select *Include All Types*.
7. To exclude instances, click *Select* to the right of the *exclude inst* field, and select the device instances that you want to exclude from the plot.
8. To include specific instances, click *Select* to the right of the *include inst* field, and select the instances on the schematic.
9. Specify the number of results you want to plot.
10. Click *Plot*.

    The total noise at the output for the top three BJTs is plotted.

To see which transistor goes with which trace, click the plus sign (highlighted in red on the previous graphic) to the left or the text at the upper left of the display area. The legends expand, and you can see which trace is which in the *Name* field on the left side

© 1994-2020 All Rights Reserved.

of the display tool. You can also either select the trace or the individual entry in the *Name* field and the trace and legend will highlight.

Viewing the exact noise mechanism within the devices

The sideband with the largest contribution was determined by selecting *Sideband Output* and plotting. The specific component was identified by selecting *Instance Output*. The specific noise mechanisms can be identified using *Source output*. This plots the largest individual noise mechanisms for the sideband you select.

© 1994-2020
All Rights Reserved.

On the *Direct Plot Form*:

© 1994-2020
All Rights Reserved.

1. Select *Source Output* from the *Function* section.
2. Select *V/sqrt(Hz)* or *v**2/Hz*.
3. Select the appropriate modifier.
4. Select the sideband number.
5. Select the type of component to display.
6. If you want every noise source to be considered, select *Include All Types*.
7. To exclude instances, click *Select* to the right of the *exclude inst* field, and select the device instances you want to exclude from the plot.
8. To include specific instances, click *Select* to the right of the *include inst* field, and select the instances on the schematic.
9. If you want to restrict to one device, put its instance name in the *include inst* field. A list of devices separated by a space is also acceptable.
10. If you want to exclude a device (or devices) enter the instance name(s) in the *exclude inst* field.
11. Specify the number of results you want to plot in the *by top* field.
12. Click *Plot.*

    The exact noise mechanism within the devices is plotted. To see which noise source goes with which trace, select the plus sign to the left or the text at the upper left of the display area. The legends will be shown individually in the *Name* field of the waveform tool. The device name and the individual noise contributor will be shown. You can select either the individual legend or the trace and both the legend and the trace will highlight.

© 1994-2020 All Rights Reserved.

For a list of the noise parameters for the different device types, see the list beginning just after the noise summary.

- The base current shot noise is in green.
- The base parasitic resistance is yellow.
- The flicker noise is red.

© 1994-2020
All Rights Reserved.

Viewing the total noise currents within the devices

© 1994-2020
All Rights Reserved.

This capability has limited value. All the noise currents from all the noise mechanisms within the instance are added up. No gain function is applied or are available for this measurement. It just adds up everything within that instance.

1. Select *Instance source*.
2. Select *A/sqrt(Hz)* or *A**2/Hz* in the Signal Level selection.
3. Select *Magnitude* or *dB20* in the *Modifier* section.
4. Select the sideband number.
5. If you want every noise source to be considered, select I*nclude All Types*.
6. To exclude instances, click *Select* to the right of the *exclude inst* field, and select the device instances you want to exclude from the plot.
7. To include specific instances, click *Select* to the right of the *include inst* field, and select the instances on the schematic.
8. Set the number of curves to plot by setting the Truncate function.
9. Click *Plot*.
10. The frequency response curves are displayed.

© 1994-2020
All Rights Reserved.

Plotting the individual noise source currents

© 1994-2020
All Rights Reserved.

This capability plots the individual noise currents within the device. it gives you the noise at the source instead of at the output of the simulation. This enables you to identify the largest individual noise sources in the circuit. Note that the largest noise source in the circuit may not contribute the largest noise at the output of the circuit. It depends on the relative gains from the individual sources to the output. These gain functions are available, and are shown in the next section.

1. Select *Primary Source*.
2. Select *A/sqrt(Hz)* or *A**2/Hz* in the *Signal Level* section.
3. Select *Magnitude* or *dB20* in the Modifier section.
4. Select the sideband number.
5. If you want every noise source to be considered, select *Include All Types*.
6. To exclude instances, click *Select* to the right of the *exclude inst* field, and select the device instances you want to exclude from the plot.
7. To include specific instances, click *Select* to the right of the *include inst.* field, and select the instances on the schematic.
8. Set the number of curves to plot by setting the Truncate function.
9. Click *Plot*.

© 1994-2020 All Rights Reserved.

10. The frequency response curves are displayed.

For a list of the noise parameter names, please see the table under the section Noise Summary on page 441.

© 1994-2020
All Rights Reserved.

Plotting the gain from the individual noise sources to the output

© 1994-2020
All Rights Reserved.

This capability gives the gain from the individual noise currents to the output of the circuit. The noise current from the previous section times the gain that is plotted in this section gives the noise at the output of the circuit.

1. Select *Source Noise Gain*.
2. Select *A/sqrt(Hz)* or *A**2/Hz* in the *Signal Level* section.
3. Select *Magnitude* or *dB20* in the *Modifier* section.
4. Select the sideband number.
5. If you want every noise source to be considered, select *Include All Types*.
6. To exclude instances, click *Select* to the right of the *exclude inst* field, and select the device instances you want to exclude from the plot.
7. To include specific instances, click *Select* to the right of the *include inst.* field, and select the instances on the schematic.
8. Set the number of curves to plot by setting the Truncate function.
9. Click *Plot*.

© 1994-2020 All Rights Reserved.

The frequency response curves are displayed. For a list of the abbreviations used for the noise parameters, please see Noise Summary on page 441.

Multiple hbnoise

Multiple hbnoise allows multiple pnoise analyses to be run after a harmonic balance simulation so that you can characterize the noise at several points in your circuit. For example, in an LNA-Mixer circuit, you might want to characterize the noise at the LNA output and at the output of the mixer. Multiple hbnoise allows these analyses to be run without needing multiple simulations.

© 1994-2020
All Rights Reserved.

Consider an LNA-Mixer circuit shown below.

In the above circuit, there are two outputs that need to be measured. The first is at the LNA output, and the second is at the mixer output. The noise frequencies are different at the two outputs.

© 1994-2020
All Rights Reserved.

First, set up the harmonic balance analysis. For more information, see the Harmonic Balance section at the beginning of this chapter.

© 1994-2020
All Rights Reserved.

In the hbnoise *Choosing Analyses* form:

1. Select *Multiple hbnoise*, and set up the noise analysis for one of the outputs. The mixer output is shown below.

© 1994-2020
All Rights Reserved.

2. Select the second line in the multiple hbnoise section and set up that analysis.

© 1994-2020
All Rights Reserved.

3. Run the simulation. When the simulation completes, select *Results - Direct plot - Main Form*. When multiple hbnoise is run, the *Direct Plot Form* adds the ability to select which hbnoise results to plot. This is the purpose of the Multiple Out selection.

© 1994-2020
All Rights Reserved.

4. Click *Plot*. The single-sideband noise figure measurement for the LNA is shown below.

© 1994-2020
All Rights Reserved.

5. To plot the noise figure of the overall circuit, set the *Multiple Out* selection to the Mixer output at Port2. For direct conversion, usually double-sideband noise figure is desired.

6. Click *Plot*.

© 1994-2020
All Rights Reserved.

The waveform tool shows the noise figure curve and the integrated noise figure to 1MHz.

Noise as a Function of Blocker Power

This is a classic desensitization measurement that measures noise figure as the blocker amplitude gets large. In order to accomplish this, variables need to be added to the RF source

© 1994-2020 All Rights Reserved.

so that the input power of the blocker can be easily swept. The RF power in dBm is set by the variable prf in the following properties list.

- Note that the source has two large-signal tones set up.

© 1994-2020
All Rights Reserved.

- The first large-signal tone has the frequency set to the variable *fb1* and the amplitude set to *prf*.
- The second tone has the frequency set to the variable *fb2* and the amplitude set to *prf*. Either of the variables can be used to set the blocker frequency.

To measure the noise as a function of blocker power:

1. Apply the signals that cause the large-signal effects in the harmonic balance (hb) analysis, and then run an hbNoise analysis.

   - The LO is set to 2.4GHz. This is highlighted in red above.
   - The first RF is set to 2.5G and -30 dBm. This is highlighted in blue above. This is the blocking signal.
   - The second RF tone is disabled by setting its frequency to zero.

© 1994-2020 All Rights Reserved.

In this case, the frequency set by `fb1` is the blocker. On the hb *Choosing Analyses* form:

© 1994-2020
All Rights Reserved.

2. Set up the hb analysis appropriately.

   Because the RF power will be quite high, the number of RF harmonics is higher than normal to preserve the accuracy.

3. Sweep the RF power over an appropriate range.

© 1994-2020 All Rights Reserved.

An appropriate range depends on what you are designing. Generally, you start about 15 to 25dB smaller than the compression point, and stop the sweep near the compression point. On the hbnoise *Choosing Analyses* form:

© 1994-2020 All Rights Reserved.

© 1994-2020
All Rights Reserved.

4. Usually, a single frequency is run in hbnoise, when the blocker power is swept. This is done to allow the direct plot functions to work properly after the simulation.
5. Specify the desired output frequency.
6. For *Maximum sideband*, either leave the field blank which causes SpectreRF to calculate all the noise mixing frequencies from all the harmonics set in the hb *Choosing Analyses* form, or specify the same number that is used to set the highest number of harmonics in the hb *Choosing Analyses* form.
7. The rest of the properties are set up as they would be in a normal simulation. Once you have finished setting up the hb and hbnoise *Choosing Analyses* forms, run the simulation.

© 1994-2020 All Rights Reserved.

8. When the simulation completes, select *Results - Direct Plot - Main Form* in ADE Explorer.

9. Select the desired noise figure measurement.

10. Click *Plot*.

© 1994-2020
All Rights Reserved.

The resulting plot appears.

11. To make the plot more useful, right-click one of the numbers on the X Axis and select *Swap Sweep Var* from the context menu.

The *Swap Sweep Var* window is displayed. Note that the *Sweep Variables* field is set to *prf*, which is the variable that was swept.

12. Specify a value in the *X Value* field. This is the frequency of the noise figure measurement and in this example it is 10MHz.

© 1994-2020
All Rights Reserved.

13. Click *OK*.

A new subwindow is displayed with the noise figure on the vertical axis, and the blocker power on the X Axis.

Select Card mode to display just the right subwindow.

Noise Type = sampled(jitter)

Driven jitter is usually used for measuring the noise in a digital circuit at a specific threshold crossing. This is usually the voltage that causes succeeding stages to be clocked.

Note: To get a correct slope of the waveform n this mode, you must manually specify enough harmonics. Auto harmonics will get you close but might not give you enough harmonics.

In the hb Choosing Analyses form:

© 1994-2020 All Rights Reserved.

1. First, set a variable for the number of harmonics.

© 1994-2020
All Rights Reserved.

2. Set the variable to a small number in ADE.
3. Run the simulation.
4. Plot the timedomain waveform with Add to Outputs selected in the Direct Plot Form.

5. Now run a sweep in ADE.

   After the simulation completes, a family of waveforms will be plotted.

© 1994-2020
All Rights Reserved.

6. Zoom in to the rising and falling edges to see the waveform details.

© 1994-2020
All Rights Reserved.

The rising edge needs more harmonics than the falling edge. Approximately 40 harmonics are needed in the example shown above.

7. Disable the sweep and set the variable to the number of harmonics you determine.

   In the above example, it was set to 40.

8. On the hbnoise *Choosing Analyses* form, set the stop frequency as half the input frequency.

9. Set the start frequency as four decades below the stop frequency.

   About five points per decade is usually sufficient.

10. Leave the *Maximum sideband* blank.

Edge Crossing Mode

In this mode, you can specify a trigger which is an edge in the circuit. This determines the time in the waveform to make the measurement. The measurement is specified separately in order to allow one waveform to set the time and measure another net at that time.

© 1994-2020 All Rights Reserved.

- Select *Enabled*.
- Set the *Threshold Value* set appropriately.
- Set *Edge Direction* to *rise* or *fall*.

© 1994-2020 All Rights Reserved.

- (Optional) Set *Sleep Time*, which is the length of time that Spectre will wait before checking for the crossing event in hbnoise sampled(jitter) analysis.

.

© 1994-2020
All Rights Reserved.

When the simulation completes, use the *Direct Plot Form* to plot the jitter.

For driven circuits, select *Jee*, which is edge-to-edge jitter. In the below example, 77p is the rising edge, and 275p is the falling edge.

© 1994-2020
All Rights Reserved.

The output noise frequency response and the edge-to-edge jitter are displayed. The jee measurement is one standard deviation for the expected jitter.

© 1994-2020
All Rights Reserved.

Sampled phase mode

First, plot the timedomain waveform for the net you want to measure. Place markers at the lowest and highest voltage you want to explore.

In this case, times from 72psec to 84psec would capture the variation in noise voltage from 1 Volt below the threshold to 1 Volt above the threshold.

On the hbnoise *Choosing Analyses* form:

© 1994-2020 All Rights Reserved.

1. Select *Sampled Phase*.

© 1994-2020
All Rights Reserved.

2. Set the stop frequency to half the input hb frequency.

3. Set the start frequency to four decades smaller than the stop frequency.
4. Leave the *Maximum sideband* field blank.
5. Select the net to be measured.
6. Type *1* (one) in the *Samples Per Period* field.
7. Select the *Add Specific Points* check box.
8. Type a list of times in the *Add Specific Points* field.
9. Click *Add*.
10. Click *OK* and run the simulation.

© 1994-2020
All Rights Reserved.

11. After the simulation completes, select *Results - Direct Plot - Main Form*.

12. If needed, select the appropriate noise analysis in the *Measurement Selection* drop-down list.

13. Select *HB Integ Output Noise* from the *Function* section.

14. Set the frequency range to the values used in hbnoise.

15. If needed, select *Add To Outputs*.

16. Click Plot.

© 1994-2020 All Rights Reserved.

The noise waveform appears in another subwindow.

17. To put the noise waveform in the same subwindow as the output waveform, right-click the X Axis in the left subwindow.

18. Select *Allow Any Units*.

19. Drag the noise waveform into the same subwindow as the waveform and delete the other subwindow.

© 1994-2020 All Rights Reserved.

20. Zoom in to the area of interest.

The rising edge and the noise voltage are plotted.

© 1994-2020
All Rights Reserved.

Edge Delay Mode

© 1994-2020
All Rights Reserved.

1. On the hbnoise Choosing Analyses form, set the stop frequency to half the PSS frequency.
2. Set the start frequency to four decades below that.
3. Select *Edge Delay*.
4. Select the net that starts the jitter measurement time.
5. Set an appropriate *Threshold Value*.
6. Select the appropriate *Edge Direction*.
7. (Optional) Set *Sleep Time*, which is the length of time that Spectre will wait before checking for the crossing event in hbnoise sampled(jitter) analysis.
8. Select the net that stops the jitter measurement.
9. Select the rising or falling edge.
10. Click *Add*.
11. Click *OK* and run the simulation.

© 1994-2020 All Rights Reserved.

12. After the simulation completes, select *Results - Direct Plot - Main Form*.

13. If you ran multiple analyses, select the appropriate hbnoise analysis in the *Measurement Selection* drop-down list.

14. Select *Jdelay* from the *Function* section.

15. Set the frequency range to the range you set in the hbnoise analysis.

16. If desired, select *Add To Outputs*.

© 1994-2020
All Rights Reserved.

17. Click *Plot*.

The jitter is shown in the marker attached to the trace.

Oscillators

Noise Type = timeaverage PM noise

Noise type=timeaverage is used to measure the USB, LSB, AM and PM noise separately. USB and LSB results have the total noise (AM and PM) and are by definition single sideband results. For oscillators, PM is suggested because usually the PM component of noise is desired. In the Direct Plot Form, you must use PM to plot phase noise. You cannot do so using the USB or LSB results.

© 1994-2020 All Rights Reserved.

1. First, set variables to define the tuning voltage (vt) and power supply voltages (VDD) in ADE Explorer.

© 1994-2020 All Rights Reserved.

2. Set up the hb form as shown in the following figure. For more information, see the hb section at the beginning of this chapter.

© 1994-2020
All Rights Reserved.

3. Next set up the hbnoise form (More information is provided below).

© 1994-2020
All Rights Reserved.

4. The sweep defaults to relative to the frequency of the specified harmonic of the hb analysis. The frequencies specified in the *Choosing Analyses* form are added to the frequency of the specified harmonic when *relative* is selected. In this case, the first harmonic is specified because there no frequency multiplier on the output of the oscillator.
5. Relative to the oscillator frequency, specify a frequency offset.
6. Either specify the number of harmonics for the hb, or leave the *Maximum sideband* field blank, which has the same effect.
7. Set *Output* to *voltage*.
8. Select the output node.
9. If you want noise separation, you must first select *sources* from the *Noise Type* drop-down list and then you can select the noise separation.
10. Run the simulation.
11. To plot the single-sideband phase noise, in ADE Explorer, select *Results - Direct Plot - Main Form*.

On the *Direct Plot Form*:

1. Select *hbnoise* from the *Analysis* section.

© 1994-2020 All Rights Reserved.

2. Select PM from the *Noise Type* section.

3. Select *Output Noise* from the *Function* section.

4. Select *dBc/Hz* from the *Units* section.

5. (Optional) To measure phase noise at specific frequencies using expressions in ADE, type a list of frequencies separated by spaces in the *Value At (Hz)* field. This requires you to click the Add To Outputs check box at the bottom of the form.

6. Select *SSB* or *DSB* from the *Noise Convention* section.

© 1994-2020 All Rights Reserved.

7. Click *Plot*.

© 1994-2020
All Rights Reserved.

Commonly Used hbnoise Options for Oscillators

For the rest of the options, please see the Commonly Used hbnoise Options.

© 1994-2020
All Rights Reserved.

Lorentzian

There are two ways of looking at phase noise. One way is to think of measuring the spectral output of the oscillator with a network analyzer. In this case, as the frequency gets close to the oscillator frequency, the amplitude starts going up, levels off, and then drops as the oscillator frequency is passed. In this case, the noise cannot be larger than the oscillations themselves, which is 0 dBc. If this is how you view phase noise, set the *lorentzian* option to *yes*.

If you think in terms of jitter, in one cycle of the oscillator output, because of the noise, you can calculate one standard deviation of the timing. If you go two cycles, and you integrate the noise with respect to time, you get twice as much noise power, or about 1.4 times the noise voltage or jitter. When you get to an infinite number of oscillator cycles, the jitter also becomes infinite. As you increase the time, you are lowering the noise frequency, and in this case, as the time becomes large, the frequency becomes very low. At infinite time, you get infinite jitter, and this occurs at zero offset from the carrier. Jitter is just another way of looking at phase noise. In other words, the phase noise can easily exceed the carrier amplitude (0 dBc) as the offset frequency becomes small because it is heading to infinity at zero frequency offset. If this is how you view phase noise, set the *lorentzian* option to *no*.

Lorentzian=yes means calculate the leveling off. *Lorentzian=cornerfreqonly* calculates the continuously rising phase noise, but it places a marker on the phase noise curve at the frequency where the phase noise would level off. *no* causes the phase noise to continue to rise as the offset frequency approaches zero.

Note: The *lorentzian* option is supported only when *Noise Type* is set to *timeaverage*.

Comparison Between Lorentzian Yes and No

As noted above, lorentzian causes leveling off of the phase noise at low frequency.

Note that when *lorentzian=no*, the phase noise curve goes above 0 dBc. This would mean that the noise is larger than the amplitude of the oscillator. This is because hbnoise is a small-signal analysis that does not recognize the large-signal limits of the circuit like clipping. It is

© 1994-2020 All Rights Reserved.

similar to AC where you can specify an input magnitude of 1 megavolt, and see 10 megavolts on the output.

To plot cycle jitter or cycle-to-cycle jitter:

© 1994-2020
All Rights Reserved.

1. On the *Direct Plot Form*, select *hbnoise* from the *Analysis* section.

2. Select *Jc* or *Jcc* from the *Function* section.

3. The frequency range will default to the range set in the *Choose Analyses* form.

   Cycle jitter (sometimes called period jitter) and cycle-to-cycle jitter are calculated by integrating the noise in the phase noise curve. Cycle jitter fixes the first threshold in time, and measures one standard deviation of the variation of the crossing of the second edge.

© 1994-2020 All Rights Reserved.

Cycle-to-cycle jitter measures one standard deviation of the variation in the period of the waveform.

4. Click *Add To outputs*.

   The Jc or Jcc calculation is sent to the ADE for evaluation. Click the Plot Outputs icon to see the result.

Noise Type = sampled(jitter)

This is usually applied to oscillators with square wave output. In most cases, pss shooting with pnoise will run faster and be less error prone. To run hb with hbnoise, ensure that you have enough harmonics to accurately capture the slew rate of the digital output signal before you run the hbnoise analysis.

In most cases, you will get the same result except at large offset frequencies compared to PM noise from timeaverage.

This analysis gives an instantaneous measurement of jitter at a threshold voltage on the timedomain waveform. This is useful for measuring the jitter of an oscillator that is driving a digital component.

© 1994-2020 All Rights Reserved.

1. First, set up the hb analysis.

2. Set a variable to define *Number of Harmonics.*

3. Set *Oversample Factor* to *2*.

© 1994-2020
All Rights Reserved.

4. Set the variable to a reasonable starting value in ADE.

   20 is used in this example.

5. Run the simulation.

6. After the simulation completes, select Results - Direct Plot - Main Form.

© 1994-2020 All Rights Reserved.

7. Select *Add To Outputs* to enter an expression into ADE.

The waveform plots.

© 1994-2020
All Rights Reserved.

8. Run a sweep in ADE to sweep the number of harmonics over a range.

9. Run the simulation.

   A family of waveforms will be plotted.

10. Open the legend to see the value for the number of harmonics.

© 1994-2020 All Rights Reserved.

11. Highlight the traces to determine an optimum number of harmonics. Determine the smallest number that causes no further increase in slope as the number of harmonics is increased. In this example, 11 harmonics was set.

12. Set the ADE variable to the value you determined above.

© 1994-2020 All Rights Reserved.

13. Next, set up the hbnoise *Choosing Analyses* form.

In the *Choosing Analyses* form:

- The stop frequency must be set to a value close to half the oscillator frequency.

© 1994-2020 All Rights Reserved.

- Set the start frequency appropriately.
- Set the sweep parameters appropriately.
- Leave the *Maximum Sideband* field blank.

  This will use all the harmonics in the hb analysis.
- Select *sampled(jitter)* from the *Noise Type* drop-down list.
- Select *Edge Crossing* for the timing event.
- Select the output net(s).
- Specify a threshold voltage for the timedomain waveform in the *Threshold Value* field.
- Specify a crossing direction in the *Edge Direction* field. In this example, *rise* is selected.
- Select the output net.
- Click *Add*.

14. Run the simulation.

15. In the ADE Explorer window, select *Results - Direct Plot - Main Form*.

16. On the *Direct Plot Form*, select *hbnoise sampled*.

17. Select *Edge Phase Noise* from the *Function* section.

© 1994-2020 All Rights Reserved.

18. Click *Plot*.

© 1994-2020
All Rights Reserved.

The phase noise will be plotted. This is single-sideband phase noise.

© 1994-2020
All Rights Reserved.

19. Now plot the cycle jitter (Jc) or the cycle-to-cycle jitter.

20. Select *Jc* from the *Function* section.

21. Set the frequency range same as you used in hbnoise.

© 1994-2020
All Rights Reserved.

22. Click *Add To Outputs*.

An expression is entered in ADE.

23. In ADE, click the *Plot Outputs* icon.

The cycle jitter is displayed in ADE.

Creating an Oscillator Macro Source

When simulating an oscillator in hb and hbnoise, the harmonics and phase noise of the oscillator can be saved to a file. This file can then be read by vsource or port to create that same waveform and noise characteristics as the oscillator originally had.

© 1994-2020 All Rights Reserved.

1. Open an oscillator circuit, as shown below.

© 1994-2020
All Rights Reserved.

2. Open ADE Explorer.

3. Open the Choosing Analyses form and select hb.
4. Specify the fundamental frequency in the *Fundamental Frequency* field.
5. Select *conservative* for Accuracy Defaults (errpreset).
6. Select *Oscillator*.

© 1994-2020 All Rights Reserved.

7. Specify the positive oscillator node in the *Oscillator node+* field. Alternatively, click *Select* and select a node from the schematic.

© 1994-2020
All Rights Reserved.

8. Select the *Generate Osc Macro Source* option.

9. Specify the output node in the *Positive Out Node* field, as shown in the figure above.

10. Specify the file name where you want to save the oscillator frequency output in the *Save File Name* field, as shown below.

11. Click Apply.

12. Select *hbnoise* in the Choosing Analysis form.

13. Specify 1 in the *Relative Harmonic* field.

14. Specify 1 in the *Start* field and *100M* in the Stop field.

15. Select *Logarithmic* from the *Sweep Type* drop-down list and specify 5 in the *Points Per Decade* field.

16. Select *voltage* from the *Output* drop-down list and specify the positive output node in the *Positive Output Node* field.

17. Select *PM* as the contribution type.

© 1994-2020 All Rights Reserved.

18. Select the *Generate Osc Macro Source* option and specify the same file name in the *Save File Name* field that you have specified in the hb *Choosing Analyses* form. This will append the phase noise information for the oscillator in the file.

© 1994-2020
All Rights Reserved.

19. Click OK and run the simulation.

After the simulation completes, select *Results - Direct Plot - Main Form* in the ADE Explorer window. The direct plot form is displayed, as shown below.

In the Direct Plot form:

1. Select *hbnoise* from the *Analysis* section.

© 1994-2020 All Rights Reserved.

2. Select *Output Noise* from the *Function* section.
3. Select *dBc/Hz* from the *Units* Section
4. Click *Plot*.

The phase noise is plotted in the waveform window, as shown below.

© 1994-2020
All Rights Reserved.

Open the *osc_macro1* file in a terminal window.

```
#FUND=2.503112e+09
#HARMS=5
#SPECTRUM
0 5.120884e-18 1.800000e+02
1 2.776595e+00 8.469083e+01
2 3.133247e-01 -1.731461e+02
3 3.756581e-02 -1.409373e+01
4 1.571884e-02 -1.352217e+02
5 4.975467e-03 -4.617804e+01
#PHASE NOISE
1.000000e+00 4.034648e+01
1.584893e+00 3.434832e+01
2.511887e+00 2.835122e+01
3.981071e+00 2.235583e+01
6.309574e+00 1.636313e+01
1.000000e+01 1.037466e+01
1.584893e+01 4.392877e+00
2.511886e+01 -1.578407e+00
3.981072e+01 -7.533281e+00
6.309573e+01 -1.346271e+01
1.000000e+02 -1.935318e+01
1.584893e+02 -2.518510e+01
2.511886e+02 -3.093156e+01
3.981072e+02 -3.655849e+01
6.309573e+02 -4.202841e+01
1.000000e+03 -4.730999e+01
1.584893e+03 -5.239470e+01
2.511886e+03 -5.731720e+01
3.981072e+03 -6.216181e+01
6.309573e+03 -6.700678e+01
1.000000e+04 -7.180947e+01
1.584893e+04 -7.643719e+01
2.511886e+04 -8.084251e+01
3.981072e+04 -8.507823e+01
6.309573e+04 -8.921152e+01
1.000000e+05 -9.328771e+01
1.584893e+05 -9.733229e+01
2.511886e+05 -1.013590e+02
3.981072e+05 -1.053756e+02
6.309573e+05 -1.093864e+02
1.000000e+06 -1.133945e+02
1.584893e+06 -1.174041e+02
2.511886e+06 -1.214201e+02
3.981072e+06 -1.254556e+02
6.309573e+06 -1.295398e+02
1.000000e+07 -1.337390e+02
1.584893e+07 -1.381817e+02
2.511886e+07 -1.430370e+02
3.981072e+07 -1.483390e+02
6.309573e+07 -1.537753e+02
1.000000e+08 -1.589007e+02
```


© 1994-2020 All Rights Reserved.

Note the harmonics information is saved at the top and phase noise information is saved at the bottom of the file.

1. Place a source in the schematic. Use a *vsource* from *analogLib*, as shown below.

© 1994-2020
All Rights Reserved.

2. Select the *vsource* from the schematic and press q to open the Edit Objects Properties Form as shown below.

3. Select *oscmacro* from the *Source type* drop-down list, as shown in the figure above.

4. Click *OK*.

© 1994-2020
All Rights Reserved.

5. In the ADE explorer window, choose *Setup - Simulation Files* and add the path to the *osc_macro1* file that you saved under the *Include Paths* section, as shown below.

6. Click *OK*.

7. Now set up the PSS shooting analysis with the frequency specified in the osc_macro1 file, as shown below.

© 1994-2020
All Rights Reserved.

© 1994-2020
All Rights Reserved.

8. Set up the pnoise analysis with the frequency range specified in the osc_macro1 file, as shown below

© 1994-2020
All Rights Reserved.

9. Click *OK* and run the simulation.

10. Plot the results using the Direct Plot Form.
    The waveform window is displayed, as shown below.

Note that the result is the same as the oscillator circuit.

© 1994-2020
All Rights Reserved.

hbSP Analysis

Overview of hbsp

Hbsp is an extension to the idea of linear S-Parameters, so that S-Parameters can be measured for circuits that translate frequency. In linear S-Parameters, a wave is launched from the input and that wave is usually partially reflected at the circuit connection, and it propagates to the output where it can be measured. Then a wave is launched at the output source, some of it is reflected back to the output, and a wave appears at the input. Using this idea, linear S-Parameters can be used effectively to characterize many linear systems.

With hbsp, a wave is launched at the input just like in S-Parameter analysis. Some of this wave is reflected back to the input, depending on the input match. Because the system is nonlinear, this wave can mix with harmonics of the periodic source that is driving the circuit to cause waves to appear at the output at many different mixing frequencies. Hbsp supports a 2-port simulation where one of the output mixing products can be measured at a time.

As an example, imagine a mixer that has a 1.1GHz RF input, and a 1GHz Local Oscillator signal. The forward wave is launched at 1.1GHz, and at the output of the mixer 1.1GHz is produced with no frequency translation. This measures the RF to IF isolation. 100MHz and 2.1GHz are the mixing products produced by mixing with the first harmonic of the LO signal. For a down conversion mixer, the 100MHz term is desired to measure the conversion gain. One of these waves can be measured in hbsp. Like linear S-Parameters, hbsp measures the power gain of the system.

For the reverse gain and match calculation, the input is at the output frequency of the mixer, and the output is at the input frequency of the mixer. The forward gain path is from 1.1GHz to 100MHz, and the reverse gain path is from 100MHz to 1.1GHz. Therefore, S11 is a measure of the input match at the RF frequency, and S22 is the output match at the IF frequency. S21 measures the forward conversion gain, and S12 measures the reverse conversion gain.

© 1994-2020 All Rights Reserved.

Example

Consider a double-balanced diode mixer, as shown below.

The properties of the LO source are as follows. The frequency and amplitude are set to variable names so they can be changed in ADE Explorer without changing the schematic.

© 1994-2020
All Rights Reserved.

© 1994-2020
All Rights Reserved.

The RF source is similar, but with different variable names.

© 1994-2020
All Rights Reserved.

The variables are set in the ADE Explorer window. The *frf* variable is set to 0 (zero) in order to disable the large-signal waveform. Only the *LO* is applied at 1GHz. 1.4 volts is +13 dBm.

© 1994-2020
All Rights Reserved.

Next, set up the hb *Choosing Analyses* form. The hb analysis is used to characterize the nonlinearity of the system and measures the amplitude of the harmonics.

© 1994-2020
All Rights Reserved.

Harmonics are set to auto so that the simulator will choose the number of harmonics automatically, and oversample factor is raised above one because the currents in the diodes are very nonsinusoidal. For more information on setting the oversample factor, refer to the harmonic balance section at the beginning of this chapter.

© 1994-2020 All Rights Reserved.

Down Conversion and RF to IF Isolation hbsp Setup

1. Open the hbsp *Choosing Analyses* form from the ADE Explorer window.

© 1994-2020
All Rights Reserved.

2. Specify the input or output frequency range in the *Start* and *Stop* fields.

   Both up conversion and the down conversion will be shown.The LO frequency is 1GHz as was shown in the harmonic balance *Choosing Analyses* form. In the figure below, the frequency range chosen is 1.05G to 1.15G. For down conversion, this is the input frequency range. The down-converted output is from 50M to 150M.

3. Enter 1 in the *Port #* column.

4. Click *Select Port* and select the input port in the circuit.
5. Click *Choose Harmonic*.

© 1994-2020
All Rights Reserved.

The Choose Harmonic window is displayed.

6. In the *Choose Harmonic* window, select the input frequency range.
7. Click *OK*.
8. Click *Add*.

   Port 1 is added to the list with the input frequency range.

9. Follow the steps from 2 through 8 to add a second IF output port at the downconverted frequency. The second port is added, as shown below.

© 1994-2020
All Rights Reserved.

10. Follow the steps from 2 through 8 to add a third IF output port at the non-converted frequency. This will measure the RF to IF isolation. The third port is added, as shown below.

The noise analysis is the same as the noise analysis in hbnoise. Refer to the hbnoise section for details. In the figure below, noise has been disabled because we will not be performing noise analysis.

11. Run the simulation.

12. When the simulation completes, select *Results - Direct Plot - Main Form* in the ADE Explorer window.

© 1994-2020 All Rights Reserved.

The *Direct Plot Form* is displayed.

13. In the *Direct Plot Form*, select *hbsp* from the *Analysis* section.

© 1994-2020
All Rights Reserved.

14. Now plot S11, S21, and S22 by clicking the buttons near the bottom of the *Direct Plot Form*.

The waveform window is displayed, as shown below.

© 1994-2020
All Rights Reserved.

Now plot the RF to IF isolation by selecting S31.

The RF to IF isolation in the ideal case is quite high at just over 121dB.

UP Conversion hbsp Setup

Remember that the LO is at 1GHz as it was before. Now we will reverse the input and output frequencies. The input is from 50M to 60M, and there are two outputs. One output is from 850M to 950M and the other output is from 1.05G to 1.06G. We will focus on the output at 1.05G to 1.06G in this example.

The frequency sweep range is unchanged at 1.05G to 1.06G.

For up-conversion, reset the hbsp *Choosing Analyses* form and follow the same steps to specify the frequency for the two ports, as shown in the following figure.

© 1994-2020
All Rights Reserved.

Note that the input and output frequencies have changed. Now the input is from 50M to 60M, and the output is from 1.05G to 1.06G. This is done using the same *Choose Harmonic* button as before.

Also, since the gain with no frequency translation from RF to IF is not made because measuring it at 100MHz to 100MHz has little value. The 50M to 60M term at the output of an up-conversion mixer is always filtered out anyway.

© 1994-2020 All Rights Reserved.

The plots of the S-Parameters are shown below.

All the calculations that are available in sp are available in hbsp. Some things like the stability factor Kf are only defined when the input frequency and the output frequency are the same. When parameters become invalid because of frequency differences between the input and output, the *Direct Plot Form* displays an error, and the parameter does not plot.

The difference between sp and hbsp is that in sp, there is no signal applied to the circuit. In hbsp, a signal is applied to the circuit. The S-Parameters are measured with that signal applied. Also, many times the input and output frequencies are different in hbsp, and by definition, linear S-Parameters cannot change frequencies.

In the example above, the signal in the harmonic balance large-signal analysis was an LO signal for a mixer, but in the example below, the signal in harmonic balance is the RF input to a power amplifier. In the next example, the input and the output are at the same frequency and the input power is swept.

© 1994-2020
All Rights Reserved.

Amplifier with Swept Input Power

Imagine an amplifier with a tiny amplitude RF input signal at 1GHz. In this case, the amplifier is very linear, and if an S-parameter measurement is done at a small offset frequency, for example, 1.01GHz, the same measurement is obtained as a linear S-Parameter measurement at 1.01GHz.

Now imagine that the input signal at 1GHz is near the compression point of the amplifier. Because the system is much more nonlinear, if a small-signal S-Parameter measurement is done at 1.01GHz, the S-Parameters would be different. Because the input signal is large, the instantaneous input impedance changes, and this changes the input match at 1.01GHz. Similarly, the other S-Parameter measurements would be affected.

Hbsp runs after hb, so it can be used to characterize the small-signal performance with the large signal present. This is not a large-signal S-parameter measurement, but it is close. If a true large-signal S-Parameter measurement is needed, use LSSP to run that simulation. Only S11, S22, S21, and S12 can be measured with the LSSP.

1. Set the hb *Choosing Analyses* form as appropriate for your amplifier, as shown below. The delta-power at high input power is small by adding the additional points to the

© 1994-2020 All Rights Reserved.

Choosing Analyses form. At high power, things change quickly, so more resolution is needed.

© 1994-2020
All Rights Reserved.

2. In the hbsp *Choosing Analyses* form, set a single frequency point, as shown below.

© 1994-2020
All Rights Reserved.

Note that both *PORT1* and *PORT2* are set to the same frequency. Use a single frequency point so the gain and stability functions return a single point in the sweep. You can use multiple points if you are familiar with the calculator.

3. Run the simulation.
4. When the simulation completes, select *Results* - *Direct Plot* - *Main Form* in the ADE Explorer window.

© 1994-2020 All Rights Reserved.

The *Direct Plot Form* is displayed.

Plotting S21

In the *Direct Plot Form*:

© 1994-2020
All Rights Reserved.

1. Select the *hbsp* radio button from the *Analysis* section.
2. Select the *SP* radio button from the *Function* section.
3. Select the *dB20* radio button from the *Modifier* section.
4. Click *S21*.

   The waveform displays *S21* as a function of input power.

Plotting Kf

© 1994-2020
All Rights Reserved.

To plot the stability factor Kf, select the *Kf* radio button in the *Function* section and click *Plot*, as shown below. For the actual equations used for the different measurements below, refer to Chapter 8, “AnalogLib Components Used in RF Simulation,”.

© 1994-2020
All Rights Reserved.

The waveform window displays Kf as a function of input power.

Plotting Noise Figure

© 1994-2020
All Rights Reserved.

To plot the Noise Figure, select the *NF* radio button in the *Function* section and click *Plot*, as shown below.

© 1994-2020
All Rights Reserved.

The waveform window displays noise figure as a function of RF input power.

Note that this is a power amplifier and not an LNA. Most LNAs have a much lower noise figure in the low RF input power region.

© 1994-2020
All Rights Reserved.

HBSP Options

Below is the HBSP Options form.

© 1994-2020 All Rights Reserved.

Relative tolerance

Leave this option at the default value.

Specifies the relative tolerance for harmonic balance-based linear solver. Default value is 1.0e-2.

Tolerance

Leave this option at the default value.

HBSP uses an iterative solver to calculate the output amplitudes. Any iterative solver needs an error tolerance to specify when to stop iterating because the solution is accurate enough. The tolerance option specifies that accuracy for HBSP. The default value is 1e-4.

Gear Order

Gear order used for small-signal integration. The default value is 2.

Solver

Leave this option at the default. The default solver is the *turbo* solver.

Hbprecond_solver

This option is only available only when harmonic balance is selected in PSS and APS is used.

The basic solver is the only solver available in standard Spectre when harmonic balance is selected. *autoset* is the default solver in harmonic balance when APS is used. This solver is faster, but occasionally stagnates. When stagnation is detected, APS automatically switches to the basic solver, and prints a message in the Spectre output window. If you have stagnation, it will save a small amount of time to set this option to *basicsolver*.

Gear_order

Do not change this option.

© 1994-2020 All Rights Reserved.

Resgmrescycle

Leave this option at the default. For the *resgmres* linear solver, there are several different options.

hbprecond_solver

This option has two settings; *basicsolver* and *autoset*. *basicsolver* is the only solver available in standard Spectre. *autoset* is the default solver when APS is used. This solver is faster. If stagnation is detected, Spectre prints a message in the Spectre output window.

Annotate

This option controls the level of detail in the Spectre output log. Selections to the left in the form have less detail in the Spectre output log. Selections to the right in the form have increasing levels of detail in the output log.

Freqaxis

HBSP displays all the results on the same frequency scale. The default is *absin* (absolute value of the input frequency) If there is a negative input frequency, *in* will show that frequency on the negative frequency axis. *out* shows the output frequency range on the X axis.

file

Outputs S-parameter data into the specified file.

datafmt

Specifies the data format of the S-parameter output file. Possible values are *spectre*, *touchstone*, and *touchstone2*. The default value is *spectre*.

datatype

Specifies the data type of the S-parameter output file. Possible values are *realimag*, *magphase*, and *dbphase*. The default value is *realimag*.

© 1994-2020 All Rights Reserved.

AdditionalParams

additionalParams is typically used for new features that are being beta tested. Entries in this field are `keyword=value` pairs.

For more information about the other options, type `spectre -h psp` at the command prompt in a Unix shell window.

© 1994-2020 All Rights Reserved.

Harmonic Balance Transfer Function Analysis (HBXF)

One way to measure conversion gain is to apply the LO and the RF signals in harmonic balance (hb) analysis and calculate the full large-signal response at each RF input frequency, and then plot the output as a function of the sweep frequency. While this works, because all the large-signal effects and many harmonics are calculated, the required time for the simulation can be significant.

The alternative is to use a small-signal approach similar to the xf analysis. The xf analysis runs considerably faster than running the transient at multiple frequencies because it ignores the large-signal effects. In circuits with frequency translation, the xf analysis is unusable because the calculation is a linear calculation based on the DC bias point. In hbxf analysis, instead of using the DC bias point, the hb solution with just the LO is used as the basis of the calculation. The large-signal analysis calculates the harmonics of the single input that causes the frequency translation to occur, (usually the LO signal or sample clock) and then hbxf is run after the hb analysis to calculate the conversion gain. The conversion gain is calculated based on the nonlinearity in the hb solution. The overall time savings can be very significant.

Hbxf requires a shift in mentality for most engineers. We are accustomed to driving an input and then measuring what comes out. In hbxf, you define an output in the circuit and the output frequency range, and then hbxf determines the input frequencies that will cause the output and it then measures the forward conversion gain from all the sources in the circuit to the output defined in the hbxf choosing analysis form.

In hbac, we apply the LO signal (or the sample clock) in the hb analysis, and then drive the input with the small-signal hbac analysis and measure the mixing products that come out. While this is very intuitive, the output display can be very confusing to read when the conversion gain near several hb harmonics is desired. Hbxf changes the focus to the output. One output frequency at a time is considered. Hbxf is a two step process. First, the LO is applied to the mixer. This is the signal that causes the frequency translation to occur. Next the input frequencies that cause the output frequency are calculated

For example, imagine a system with a 1GHz LO signal, and a 1MHz output signal. In HBXF, we specify 1MHz as the output frequency. In this system, inputs at 1MHz, 999MHz, 1001MHz, 1999MHz, 2001MHz, and so on will cause 1MHz to be produced at the output. Next, the forward conversion gain from the sources in the circuit at all those input frequencies is calculated. Since the input frequencies are different, the conversion gain is different for every input frequency. Using hbxf, you can measure the conversion gain near a number of LO harmonics (or clock harmonics) at the same time.

© 1994-2020 All Rights Reserved.

Example: Conversion Gain (Down Conversion)

Consider a double-balanced diode mixer shown below.

© 1994-2020
All Rights Reserved.

The RF input source has the frequency and amplitude set to variable names in order to allow changing the test setup without needing to change the circuit. This is suggested for all the sources used in the circuit that might need to be changed. The property list is shown below.

© 1994-2020 All Rights Reserved.

© 1994-2020
All Rights Reserved.

The easiest way to get the variables list into the ADE Explorer window is to select *Variables -> Copy From Cellview*. To set the values, select the value field to the right of the variable, type the desired value, and press <Enter>.

The LO frequency is 1GHz and its amplitude is 1.4 volts peak. This is +13 dBm. The RF frequency is 1.1GHz, but it is disabled because the amplitude is set to 0 (zero). Setting either the frequency or the amplitude to zero disables the production of that waveform from the port.

© 1994-2020
All Rights Reserved.

The hb analysis needs to be set up first. The example below shows a single input at 1GHz. For more information, see the harmonic balance section.

© 1994-2020
All Rights Reserved.

Next, the hbxf analysis needs to be set up. Like xf, the frequency range specified in the form is the output frequency range. Maximum sideband should be left blank so that all the mixing products from all the harmonic balance harmonics are present in the output of hbxf.

© 1994-2020
All Rights Reserved.

Once the analyses are set up, the simulation can be run. When the simulation finishes, select *Results - Direct Plot - Main Form* in the ADE Explorer window.

HBXF calculates the transfer functions from all the independent sources in the circuit to that output node. Therefore, to plot, you select the source in the circuit that you want the forward

© 1994-2020
All Rights Reserved.

conversion transfer functions to be displayed. When the source in the schematic is selected, the waveform is displayed.

Note that all the short line segments are the forward conversion gain from the selected source (the input source in this case) to the output.

To measure the conversion gain, zoom in to the curve at your input frequency and place a marker to measure the gain.

© 1994-2020
All Rights Reserved.

Overview of Simulation Capabilities

The primary application of hbxf is to measure conversion gain. Hbxf is a small-signal analysis. It calculates the small-signal conversion gain. If you need a large-signal measurement of conversion gain, then use a large-signal analysis like harmonic balance (hb) or quasi-periodic steady-state (qpss) analysis. Hbxf measures gain in a way that requires some thought for most engineers who are accustomed to connecting an input, driving the system, and measuring what comes out. This is the idea of hbac, where we capture the large-signal operating point in hb, and then drive a small-signal input, and measure the mixing products that are produced. If you think of this methodology, to measure the conversion gain around several harmonics usually requires a separate simulation near each harmonic.

Hbxf was created to allow the measurement of the conversion gain transfer functions near all the hb harmonics in the same simulation run. In hbac, we have one input frequency at a time, and output mixing products at multiple output frequencies are calculated. In hbxf, we focus on one output mixing product at a time, and measure the input frequencies that cause that output frequency, and measure the forward conversion gain from the sources in the circuit at all those input frequencies to the single output frequency. Multiple output frequencies are usually analyzed in an hbxf analysis. Unlike hbac where the output can be seen at every net in the circuit, hbxf calculates the gain only from all the independent sources in the circuit. Therefore, to measure a conversion gain, you select the source instance from the *Direct Plot Form* instead of selecting a net.

For example, imagine a mixer with an LO frequency of 1GHz. Now, imagine the down conversion case where the output is 1MHz. Given the LO frequency, inputs at 1MHz, 999MHz, 1.001GHz, 1.999GHz, 2.001GHz, 2.999GHz, 3.01GHz, and so on cause an output at 1MHz. Depending on the circuit, all those different input to output paths have different gain from each input frequency to the output at 1MHz. In hbxf, you specify the highest hb harmonic to calculate the transfer functions from in the maximum sideband field in the *Choosing Analyses* form. All these different input frequencies are called sidebands in hbxf.

Hbxf has two different modes, which are set by the selection in the *Specialized Analysis* section of the *Choosing Analyses* form. *Specialized Analysis = None* measures the conversion gain that occurs given the nonlinearity created by large-signal in the circuit. *Sampled* supplies a measurement of the instantaneous gain when the gain is repeatedly measured at a single phase (point in time) of the periodic large signal. This is useful for measuring deterministic jitter in a digital circuit. Note that shooting is more suited to this measurement because many harmonics would be needed in harmonic balance to get a close approximation of the digital waveform.

© 1994-2020 All Rights Reserved.

Normal Conversion Gain Measurement (Specialized Analysis = None)

This mode is used to make a normal conversion gain measurement.

The figure above illustrates a down-conversion case where the output is just above zero Hz.

In the figure above, the top spectrum in black represents the harmonics created by the hb large-signal analysis. The negative and positive frequencies are shown and the DC component is in the center. The colored arrows show the transfer functions from the different input frequencies to the output frequency that is shown on the bottom axis. Hbxf takes the maximum sideband property from the hbxf *Choosing Analyses* form, and calculates the specified sidebands, which are input to output conversion gain transfer functions. In this example, maximum sideband is set to three. If the maximum sidebands field is left blank, then the transfer functions from all the hb harmonics are calculated.

The sideband numbers indicate which harmonic number is being mixed with for the transfer function calculation. Zero means that there is no frequency translation. One is the mixing product that occurs with an input frequency shift of +1 times the hb first harmonic frequency. Mixing frequencies to the left have negative numbers because the input frequency that causes the output is more negative than the output frequency.

For example, imagine a large-signal input at 1GHz in the hb analysis and an output frequency of 100MHz to be calculated. With no mixing, there is a gain from the sources in the circuit to the output of the circuit at 100MHz. This is called the zero sideband because there is zero frequency shift from the input to the output. This is not the same measurement as a

© 1994-2020 All Rights Reserved.

measurement in the ac analysis because the ac only includes the DC operating point, and the hbxf analysis takes into account the circuit with the LO or clock signal applied.

The input frequencies that mix with the first harmonic at 1GHz to provide an output at 100MHz are at -900MHz and 1.1GHz. The frequency that is below the zero sideband is the -1 sideband and the frequency above the zero sideband is the +1 sideband.

Similarly, there are terms that mix with the second and third harmonics.

Usually, we think of positive frequencies only. This is shown above. The negative frequency inputs are reflected up into the positive frequencies. This is the default for hbxf.

Your application sets the number of sidebands that you need. If you have specs through the fifth harmonic, you need to set maximum sidebands to 5.

Frequency Sweep

This analysis is like xf, where the **output** frequency is varied. The output frequency range to be analyzed is specified in the hbxf *Choosing Analyses* form. Hbxf then calculates the conversion gain terms.

In addition to the frequency, there is a selection called *Sweep Type*. The choices are *relative* and *absolute*. When *absolute* is selected, the frequency range is used directly with no

© 1994-2020 All Rights Reserved.

frequency conversion. When *relative* is selected, a *Relative Harmonic* field appears. The frequency sweep is shifted by the specified frequency of the harmonic in the hb analysis. For example, assume that the hb is at 1GHz, and a log sweep is desired above the first harmonic of the hb analysis. In this case, you need to select *relative* sweep, and specify 1 in the *Relative Harmonic* field. Next, type in 1K to 100M for the frequency range with a log sweep and 3 to 5 points per decade.

The sweep will start at 1G + 1K, and sweep to 1G + 100M using log spacing from 1K to 100M. This is illustrated below.

© 1994-2020
All Rights Reserved.

To use a log sweep below the first harmonic, use the same frequency range, and type in -1 as the relative harmonic. This is illustrated below.

Maximum Sideband

Sideband is the name of the hbxf conversion gain terms. The sideband number is the hb harmonic number that is mixed with. If the sideband number is negative, the input frequency is more negative than the output frequency. If the sideband is positive, the input frequency is more positive than the output frequency.

Conceptually, an infinite number of conversion gain terms exist when a single output frequency is analyzed in a nonlinear system. From the practical point of view, usually only a small number of conversion gain terms need to be measured near the LO frequency and a small number of harmonics.

There is a property in the *Choosing Analyses* form called *Maximum sideband* that is used to define how many mixing products should be calculated. If *Maximum sideband* is 0, the hbxf analysis will only contain the frequency that appears without mixing or aliasing of any type. This is not equivalent to a linear AC analysis because hbxf takes into account the

© 1994-2020 All Rights Reserved.

instantaneously varying nature of the LO or sample clock signal because of the large signal being present in the hb analysis.

When *Maximum sideband* is 1, the inputs without frequency translation and the inputs that mix or alias with the first harmonic of the hb analysis are present in the hbxf analysis. When *Maximum sideband* is 10, the hbxf analysis includes mixing through the 10th harmonic of the hb analysis. The number of sidebands can be any number up to the number of harmonics specified in the hb analysis.

Setting Harmonics and Sidebands

With harmonic balance, only the harmonics specified in the hb *Choosing Analyses* form exist in the solution. In the hbxf analysis, because only the harmonics specified in the hb *Choosing Analyses* form exist, the maximum sideband term can only be set to the number of harmonics specified in the hb analysis or less.

Note that with harmonic balance, the number of harmonics (and oversample factor if you have non-sinusoidal waveforms) in the hb analysis needs to be determined. Start with a number of harmonics in the hb analysis based on an estimate of how many harmonics the circuit actually produces with your input waveform, and run the hb and hbxf analyses. Remember that for non-sinusoidal waveforms, oversample factor also needs to be set greater than one. For more information, see the hb section at the beginning of this chapter. Run the analysis and make the desired measurement. Now increase the number of harmonics by roughly 50% and run the simulation again. If the answer does not change, then you had enough harmonics originally, and you might actually be able to decrease the number of harmonics from the original setting. If you have non-sinusoidal waveforms, increase the oversample factor to between two and four before increasing the number of harmonics to find the minimum value for both oversample factor and harmonics.

© 1994-2020 All Rights Reserved.

Hbxf is usually preferred to hbac when you need to measure both the passband and the image. Below is a plot of the hbxf and hbac output for a mixer with a 1.9GHz LO and a 100MHz IF. The passband is at 1.8GHz and the image is at 2GHz.

On the left is an hbxf plot when the input port is probed, thus measuring the gain from the input to the output of the mixer. To read the gain, go to the **input** frequency, and read the gain on the vertical scale. At the passband frequency of 1.8GHz the gain is about 5dB. The LO frequency is at the junction of the red and blue traces.

To measure passband and image gain in pac, the input frequency starts at 1.6GHz and stops at 2.2GHz. At 1.6GHz, the output frequency is -300MHz, but hbac reflects the negative frequencies to the positive frequency domain by default. When the input frequency is 1.9GHz and the output is zero Hz. As the input continues above 1.9GHz, the output frequency becomes positive, and follows the curve back to 300MHz. Most people find the curve from hbac to be more confusing than the curve from hbxf, so if you need to measure both the passband and the image, or if you need to measure the conversion gain near several of the hb harmonics, hbxf is recommended.

Note that in the hbac result above, you cannot just add the gain values at a specific frequency. If the frequency was 100MHz, you do have two different gain values of about +5dB.

© 1994-2020 All Rights Reserved.

Hbxf can measure the gain near many harmonics at the same time by increasing the maximum sideband parameter in the *Choosing Analyses* form up to the number of harmonics calculated in the hb analysis. Below is an example of conversion gain through the fifth harmonic. Your application will determine the number of sidebands to set.

Note that the conversion gain is still relatively high near the third harmonic.

Hbxf can also be used to measure the conversion gain from the battery that supplies power to the output. All the gain and conversion gain terms should be below unity, or a potential

© 1994-2020 All Rights Reserved.

oscillator is created. In the plot below, the gain from the bias source to the output at the IF frequency of 100MHz is below 0dB. This circuit is not a potential oscillator.

© 1994-2020
All Rights Reserved.

Hbxf is useful for up-conversion applications as well, with the limitation that a small-signal gain measurement is made. This is shown below.

The difference is that the output frequency is just above the first harmonic. The rules for the sidebands are the same. -1 means that the input frequency is shifted down from the output frequency by -1 * hb first harmonic frequency.

© 1994-2020
All Rights Reserved.

Usually, we think in terms of the positive frequency domain, which is shown below.

© 1994-2020
All Rights Reserved.

Sampled HBXF

This analysis is seldom used.

The basic idea is to measure the conversion gain as if the conversion gain was repeatedly measured at the same phase (instant in time) in the periodic large signal that is driving the circuit over an infinite number of cycles of the large signal. This measurement is impossible to duplicate in the lab. The primary application is for measuring the gain from the power supply to the output which is driving a digital circuit or a comparator that is triggered at a specific voltage level in order to measure deterministic jitter. It is seldom used because shooting pss with pxf is better suited for this application.

ADE Implementation

This section jumps from the theoretical to the practical with examples of how to use the individual analysis types we just covered. The focus is only on the hbxf Choosing Analyses form so you can see where the individual settings are made. For examples with all the steps shown, see the examples section that follows this section.

© 1994-2020 All Rights Reserved.

HBXF for a Normal Conversion Gain Measurement

Here is an example of the *Choosing Analyses* form for a normal conversion gain measurement.

- *Sweeptype* can be relative or absolute.

© 1994-2020
All Rights Reserved.

- The frequency sweep range is always an output frequency range. It is not a sweep in the traditional sense of connecting an input, sweeping over a frequency range, and measuring the output. Instead, it is an output frequency range to be analyzed. At each output frequency in the sweep, the input frequencies that will mix with the harmonic balance harmonics to cause that output frequency are calculated, and the forward conversion gain is calculated.
- Linear and log spacings are provided. Generally, Auto should be avoided because it will always calculate 100 frequency points. Usually, a much smaller number is needed, so it is recommended that you set linear or log, and control the resolution yourself.
- Maximum sideband is typically left blank so that all the mixing products from all the harmonic balance harmonics are present in the output of hbxf.
- The output can measure either a voltage or a current. The voltage measurement is the default. Specify the output node or nodes in the circuit. If the output is a current select probe for the output, add an iprobe from analogLib (a current probe) in series with the output in the schematic, and specify the instance name in the *Choosing Analyses* form.
- When *Specialized Analysis* is set to *None*, a normal conversion gain measurement is made.

Sampled HBXF

Sampled hbxf is typically used to measure deterministic jitter of a digital circuit caused by ripple on a power supply. Although the capability is provided in hbxf, this application is much better suited to pss with sampled pxf. Usually, hb would require many harmonics to get an accurate reproduction of the digital waveform. In this case, pss with sampled pxf is likely to be much faster. Because of this, sampled hbxf is not covered in this document. See Chapter 4 for shooting pss with sampled pxf.

© 1994-2020 All Rights Reserved.

HBXF Options

The Harmonic Balance XF Options form is shown below.

The following options are occasionally used:

© 1994-2020 All Rights Reserved.

relative Tolerance

Leave this option at the default value.

Relative tolerance for harmonic balance-based linear solver. Default value is 1.0e-2.

tolerance

Leave this option at the default value.

HBXF uses an iterative solver to calculate the output amplitudes. Any iterative solver needs an error tolerance to specify when to stop iterating because the solution is accurate enough, and the tolerance option specifies that accuracy for hbxf. For driven circuits, the default is 1e-6. For oscillators, the default is 1e-4.

gear_order

Do not change this option.

solver

Leave this option at the default.

The solver option is used only for driven circuits. The default solver is the turbo solver.

resgmrescycle

Leave this option at the default. For the resgmres linear solver, there are several different options.

hbprecond_solver

This option has two settings; *basicsolver* and *autoset*. Basicsolver is the only solver available in standard Spectre. `autoset` is the default solver when APS is used. This solver is faster. If stagnation is detected, Spectre prints a message in the Spectre output window.

© 1994-2020 All Rights Reserved.

annotate

This option controls the level of detail in the Spectre output log. Selections to the left in the form produce less detail. Selections to the right in the form have increasing levels of detail. The default value is *status*.

stimuli

This function is not currently implemented.

freqaxis

freqaxis specifies whether you want to see the negative frequency axis or not. The analysis calculates the input frequencies and forward conversion gain, so *in* and *absin* are reasonable choices. *in* displays the negative frequency axis. *absin* (absolute value of the input) displays positive input frequencies. This is the default.

save

This function is not currently implemented.

additionalParams

This is a field for `<option_name>=value` statements. Multiple `<option_name>=value` statements are allowed with a space between them. Generally, this is used to unlock the beta features.

For more information about the other options, type spectre -h hbxf at the command prompt in a Unix shell window.

© 1994-2020 All Rights Reserved.

Examples for Driven Circuits

HBXF Normal Conversion Gain Measurement

Consider the following circuit:

The circuit is a double-balanced down-conversion mixer with a 1.9GHz LO.

© 1994-2020
All Rights Reserved.

ADE Explorer has the variables set to appropriate values, as shown below.

This setup causes the LO to be driven, and both RF inputs to be turned off for the hb analysis by setting the RF frequencies to zero. You can disable a signal for the hb analysis by setting its frequency to zero.

Using HBXF Analysis for Conversion Gain Measurement

Set up the hb analysis with just the large signal that causes the frequency translation to occur.

© 1994-2020
All Rights Reserved.

For more information, see the harmonic balance section at the beginning of this chapter.

© 1994-2020
All Rights Reserved.

Set up the hbxf Choosing Analyses form, as follows:

© 1994-2020
All Rights Reserved.

1. Specify an appropriate output frequency range for your circuit.
2. You can set either linear or log spacing setting. Auto usually takes more frequency point than are required, and thus extends the runtime.
3. Leave the *Maximum Sideband* field left blank in which case, all the mixing products are calculated for all the harmonics in the hb analysis.
4. Set the output nodes, as shown in the example. If the *Negative Output Node* field is left blank, as shown above, the global ground node is used by the simulator for the reference node. If you have an output current, add an iprobe from analogLib in series with the output, and select *probe* in the *Output* section. Then specify the iprobe in the *Output Probe Instance* field.
5. Run the simulation

© 1994-2020 All Rights Reserved.

After the simulation completes, select *Results - Direct Plot - Main Form*. The *Direct Plot Form* is displayed.

In the *Direct Plot Form*:

1. Select *hbxf* from the *Analysis* section.
2. Select *db20* from the *Modifier* section.
3. Select the voltage or current source from the schematic.

   Note that when you plot hbxf results, you are selecting an instance, not a node. You will get the conversion gain results for the source you select in the circuit. The example below shows the conversion gain from the input source in the circuit.

© 1994-2020 All Rights Reserved.

The waveform window is displayed, as shown below.

Since the *Sweeptype* was set to *Logarithmic*, the data is displayed with a log axis in the waveform tool. In some cases, a linear X Axis is easier to read. To accomplish this, move the

© 1994-2020
All Rights Reserved.

mouse cursor over one of the numbers on the X Axis and click the right mouse button. Deselect *Log Scale* from the context menu. This is shown below.

To read the conversion gain, move to the desired input frequency, and read the gain off the Y axis. The gain at 1.91GHz is about 15dB. The place to look is in the red rectangle above. Alternatively, position your mouse cursor on the trace at the input frequency, and read the value in the cursor readout. You can also place a marker at the input frequency to read out the conversion gain. To do this, move your mouse cursor near the trace at the input frequency, and type m.

© 1994-2020 All Rights Reserved.

The conversion gain near the third and fifth harmonics of the LO is also reasonably high as well. The gain near the third harmonic is near -5dB, and the gain near the third harmonic is near -25 dB. See the red and blue boxes below.

© 1994-2020
All Rights Reserved.

In the Direct Plot Form, select *New Win* from the *Plotting* mode drop-down list and select the DC source that is supplying power to the circuit.

Note that the gain is about +4.5dB at 150MHz. This circuit is a potential oscillator.

Sampled HBXF Analysis

As noted earlier, this analysis is likely to run much faster using shooting pss and sampled pxf. See Chapter 4 for directions for this capability.

© 1994-2020
All Rights Reserved.

Examples for Oscillators

Conversion Gain From Power Supply to the Output Frequency

Consider the following circuit:

On the *Choosing Analyses* form:

© 1994-2020
All Rights Reserved.

1. Set up the hb analysis, as shown below. For more information, refer to harmonic balance section at the beginning of this chapter.

© 1994-2020
All Rights Reserved.

2. Now, set up the hbxf analysis, as shown below.

© 1994-2020
All Rights Reserved.

3. Set the frequency range to a frequency range that is relevant for your needs. In this example, the frequency range is from 100Hz to 100MHz above the first harmonic of the oscillator frequency.
4. Select *Logarithmic* from the *Sweep Type* drop-down list.
5. Specify 3 to 5 in the *Points Per Decade* field. Here, 4 is specified.
6. Leave the *Maximum Sidebands* field blank.
7. Specify the output node in the *Positive Output Node* field.
8. Click *OK* and run the simulation.

© 1994-2020 All Rights Reserved.

When the simulation completes, select *Results - Direct Plot - Main Form*. The Direct Plot Form is displayed, as shown below.

In the *Direct Plot Form*:

1. Select hbxf from the Analysis section.
2. Select sideband from the Sweep section
3. Select db20 from the Modifier section.

© 1994-2020
All Rights Reserved.

4. Select your input frequency range in the list in the *Input Sideband* section.
5. Select the power supply from the schematic window.

The waveform is displayed, as shown below.

To read the gain, go to the input frequency, and read the gain from the Y axis. The gain is about 57dB at 100KHz in this example.

© 1994-2020
All Rights Reserved.

Harmonic Balance Stability Analysis (HBSTB)

The HBSTB analysis is used to evaluate the stability of a periodically varying feedback circuit. It is a small-signal analysis like STB analysis, except that the circuit is first linearized about a periodically varying operating point as opposed to a simple DC operating point.

Linearizing about a periodically time-varying operating point allows the stability evaluation to include the effect of the time-varying operating point.

Evaluating the stability of a periodically varying circuit is a two-step process. In the first step, the small stimulus is ignored and HB analysis is used to compute the periodic steady-state response of the circuit to a possibly large periodic stimulus. As part of the HB analysis, the periodically time-varying representation of the circuit is computed and saved for later use. In the second step, a probe is used to compute the loop gain of the zero sideband. The local stability can be evaluated using gain margin, phase margin, or a Nyquist plot of the loop gain.

© 1994-2020 All Rights Reserved.

Example: Variation in Loop Gain of an Opamp with Swept Input Signal Amplitude

Consider the opamp circuit below.

To use hbstb, an `iprobe` from analogLib (single-ended) or a `diffstbprobe` from analogLib (differential) needs to be added in series with the feedback path. Note the arrow in the current probe is in the same direction as the direction of the feedback.

The input source to the circuit has the frequency and amplitude set to variable names in order to allow changing the test setup without needing to change the circuit. This is suggested for all the sources used in the circuit that might need to be changed. This allows changes in frequency or amplitude from the ADE Explorer environment without needing to change the schematic itself.

© 1994-2020 All Rights Reserved.

The property list for the input source is shown below.

The easiest way to get the variables list into ADE Explorer is to select *Variables -> Copy From Cellview*. To set the values, select the value field to the right of the variable, type in the desired value, and press <`Enter`>. When you have set the variables, if you want to add the

© 1994-2020
All Rights Reserved.

values back into the cellview, in the ADE Explorer window, select *Variables - Copy To Cellview*, and then perform a *Check and Save* in the schematic.

The input frequency for this example is 100KHz and its amplitude is 100 millivolts peak.

The basic strategy is to apply the signal that causes the distortion in the hb analysis, and then measure the loop gain using the hbstb analysis.

The hb analysis needs to be set up first. The example below shows a single input at 100KHz. The amplitude is swept from 0.1 volts to 1.5 volts peak with a step value of 0.2 volts. For more information, see the hb section at the beginning of this chapter. This will characterize the

© 1994-2020 All Rights Reserved.

amplifier at a range of different input amplitudes. At each amplitude, both the hb and hbstb analyses are run.

© 1994-2020
All Rights Reserved.

Next, the hbstb analysis needs to be set up. The frequency range depends on your application. The *Probe Instance* is the `iprobe` from analogLib in the circuit. If you have a differential circuit, use `diffstbprobe` from analogLib. `diffstbprobe` allows the measurement of common-mode and differential-mode loop gain.

Once the analyses are set up, run the simulation. When the simulation finishes, select

Results - Direct Plot - Main Form in the ADE Explorer window.

© 1994-2020
All Rights Reserved.

The example below shows the *Direct Plot Form*. If you want dB for the loop gain amplitude plot, plot the magnitude and phase separately This example shows dB for the loop gain magnitude plot.

Here is the output voltage of the opamp versus input voltage for the sweep. The curve begins to compress at 1.1 volts peak.

© 1994-2020
All Rights Reserved.

© 1994-2020
All Rights Reserved.

Here is the loop gain and phase curve for the 0.1 volt input.

Note that the DC phase is 180, and not zero. From feedback theory, we would expect zero. So what causes this difference? In feedback theory, we always assume there is a forward and reverse transfer function, and an inverting summing junction with a gain of unity. In feedback theory, we plot the forward and reverse transfer functions only. When the phase is zero, and then applied through the inverting input, the phase becomes 180. In the circuit, we cannot separate out the summing junction. Thus, the condition of instability is a linear magnitude of 1 and a phase of zero, or multiples of 360 degrees.

Vertical marker V2 has been placed at (nearly) unity gain (0dB). The phase is about 87.8 degrees. The distance to zero phase is 87.8 degrees. This is the phase margin for this circuit at low input amplitude. The unity gain frequency is about 800KHz.

© 1994-2020
All Rights Reserved.

Below is the plot with a 1.1 volt input.

The DC gain has dropped from about 66dB to about 52 dB. The unity gain frequency has dropped from about 800KHz to about 178KHz. The phase margin has actually improved.

Overview of Simulation Capabilities

The HBSTB analysis is used to evaluate the stability of a periodically varying feedback circuit. First the large signal is applied in HB. An `iprobe` (single-ended) or a `diffstbprobe` (differential) is placed in the circuit in series with the feedback path. Both probes are in analogLib. HBSTB applies a voltage to the source(s) in the probe, and measures the amplitude and phase of the signal across the probe. This is the loop gain of the circuit. `diffstbprobe` allows the measurement of common-mode and differential-mode gain in the differential circuit.

In some cases, placing the probe in the circuit can break some local feedback paths. If this happens, the loop gain calculation will be incorrect. If you are unsure if this could be happening in your circuit, place the probe in a different position in the loop. If the loop gain

© 1994-2020 All Rights Reserved.

stays the same, both results are correct. If it changed, try a third point in the circuit. When the two points agree, you have the loop gain.

Unlike shooting pss with pstb which requires maxacfreq to be set in pss to go to a very large stb frequency compared to the pss frequency, hbstb just needs enough harmonics to accurately characterize the circuit. If the highest frequency in hbstb is larger than the frequency of the highest harmonic in hb, make sure you compare the hbstb result with a run that includes more harmonics in hb. If the result didn't change in hbstb, the original number of harmonics was enough. If it did change, you need more harmonics in hb.

© 1994-2020 All Rights Reserved.

ADE Implementation

Here is an example of the Choosing Analyses form for a normal loop gain measurement.

- The frequency sweep range specifies the limits and spacing of the loop gain measurement.

© 1994-2020
All Rights Reserved.

- Linear and log sweeps are provided. Generally, log sweeps are used. The number of points depends on how rapidly the loop gain changes with frequency in your circuit. High Q circuits will need more resolution than shown in this example.
- A probe must be used in the circuit and specified in the *Probe Instance* field. For single-ended circuits, an `iprobe` from analogLib should be used. For differential circuits, a `diffstbprobe` should be used. When `diffstbprobe` is used, you will have an additional choice in the *Choosing Analyses* form that allows the selection of differential or common-mode measurements of loop gain.

© 1994-2020 All Rights Reserved.

HBSTB Options

The HBSTB Options form is displayed below.

relative tolerance

Leave this option at the default value.

© 1994-2020 All Rights Reserved.

Specifies the relative tolerance for harmonic balance-based linear solver. Default value is 1.0e-2.

tolerance

Leave this option at the default value.

HBSTB uses an iterative solver to calculate the output amplitudes. Any iterative solver needs an error tolerance to specify when to stop iterating because the solution is accurate enough, and the tolerance option specifies that accuracy for hbstb. The default value is 1.0e-04 for harmonic balance-based solver.

gear_order

Do not change this option.

solver

Leave this option at the default.

The solver option is used only for driven circuits. The default solver is the turbo solver.

regmrescycle

Leave this option at the default. For the resgmres linear solver, there are several different options.

hbprecond_solver

This option has two settings; `basicsolver` and `autoset`. *basicsolver* is the only solver available in standard Spectre. *autoset* is the default solver when APS is used. This solver is faster. If stagnation is detected, Spectre prints a message in the Spectre output window.

oscsolver

Leave this option at the default.

The *oscsolver* option is used only for oscillators. The default solver is the turbo solver.

© 1994-2020 All Rights Reserved.

annotate

This option controls the level of detail in the Spectre output log. Selections to the left in the form produce less detail. Selections to the right in the form have increasing levels of detail. The default value is *status*.

save

This function is not currently implemented.

nestlvl

This function is not currently implemented.

additionalParams

This is a field for `<option_name>=value` statements. Multiple `<option_name>=value` statements are allowed with a space between them. Generally, this is used to unlock the beta features.

For more information about the other options, type spectre -h hbstb at the command prompt in a Unix shell window.

© 1994-2020 All Rights Reserved.

Oscillator Example

For the differential oscillator circuit below, a `diffstbprobe` has been placed at mirror locations in the differential circuit. This is shown in the red box below.

© 1994-2020
All Rights Reserved.

1. First, an hb analysis needs to be set up. For more information, refer to the harmonic balance section at the beginning of this chapter.

© 1994-2020
All Rights Reserved.

2. Now set up the hbstb analysis. The resolution of this sweep is quite high.

The frequency sweep range, and the number of points will be determined by your circuit. In this case, from linear stb analysis, all the interesting variations started at 1GHz. Narrow peaks were detected, which forced many points in order to resolve the amplitude of the peaks.

3. Click *OK* and run the simulation.

© 1994-2020
All Rights Reserved.

When the simulation completes, select *Results - Direct Plot - Main Form*. The *Direct Plot Form* is displayed, as shown below.

Now the loop gain and phase are plotted using the *Direct Plot Form*.

© 1994-2020
All Rights Reserved.

The plot of the loop gain and phase is shown. The red trace is loop gain, and the blue trace is the phase.

A vertical marker has been placed at the zero degree point on the phase curve. The frequency readout shows 5.150GHz.

© 1994-2020
All Rights Reserved.

The plot of the output spectrum is shown below.

Note that the frequency in hb is the same as the small-signal hbstb analysis. The nonlinearities of the oscillator are taken into account in the hbstb simulation.

© 1994-2020
All Rights Reserved.

4

Single Input Large and Small-signal Analyses

Overview of Simulation Capabilities

SpectreRF offers two different modes, Spectre and APS. Spectre uses traditional simulation techniques. APS makes the CMOS models more efficient and allows multi-threading for the device current evaluation. In addition, APS adds full parallel solving of all the matrices, thus speeding up the simulation even more. There is no loss in accuracy because there is no change in the actual equations that are simulated, or in the options that control the accuracy of the simulation. These choices are available and are implemented for periodic steady-state (pss) shooting and pss harmonic balance. Both large-signal and small-signal `pac`, `pxf`, `psp`, `pstb`, and `pnoise` analyses have the Spectre and APS implementations.

Overview of Periodic Steady-State (pss) Analysis

Pss calculates the settled time and frequency domain response of your system. Harmonic balance and shooting engines are available. Harmonic balance solves in the frequency domain and uses an ifft to calculate the time-domain response. Shooting calculates the settled time-domain waveforms and calculates the frequency-domain content using a Fourier transform.

All the harmonics of the beat frequency are calculated in the pss analysis up to the maximum harmonic specified in the *Choosing Analyses* form.

Pss needs to simulate an integer number of cycles. This limits the practical application of pss to single-input frequency circuits, or multiple-input frequency circuits that have a relatively high common multiple frequency, called the beat frequency. For example, if 1GHz and 1.1GHz frequencies are applied, the beat frequency is 100M. Solving for the fifth harmonic of the 1.1GHz input requires that 55 harmonics be calculated. This is a reasonable problem for pss. If the inputs are 1GHz and 1.001GHz, 5005 harmonics are needed to calculate through the fifth harmonic of 1.001GHz. This is not a good application for pss. If you need to simulate this kind of system, use qpss-harmonic balance or hb analysis instead.

© 1994-2020 All Rights Reserved.

Example

A behavioral amplifier circuit is shown below.

© 1994-2020
All Rights Reserved.

The properties for the amplifier are shown below.

© 1994-2020
All Rights Reserved.

The amplifier has input and output impedances of 50 ohms and the gain is 20dB.

The input source on the left of the schematic is called a port. The properties of the port are shown below.

© 1994-2020
All Rights Reserved.

The port is set to generate sinusoids, and one signal is defined. For the pss analysis, you must enter a name for the signal in the *Frequency name 1* field. Variable names are used to set the frequency and amplitude so that the frequency and amplitude can be changed in ADE Explorer without needing to change the schematic. The ADE Explorer setup is shown below.

The input frequency is 2.4GHz and the input power is -30 dBm.

Next, the pss analysis is set up, as shown below.

© 1994-2020
All Rights Reserved.

© 1994-2020
All Rights Reserved.

The pss *Choosing* Analyses form is set to a single input at 2.4GHz with 6 harmonics and moderate accuracy.

The simulation is run, and then in ADE Explorer, *Results* - *Direct Plot* - *Main Form* is selected. The *Direct Plot Form* is displayed, as shown below.

© 1994-2020
All Rights Reserved.

Here, dBm is calculated based on the voltage on a node. At the bottom of the *Direct Plot Form* are directions. In this case, the net on the schematic needs to be selected.

The waveform window is shown below.

The basic display shows the harmonics that were calculated by pss. Markers have been placed on the first harmonics of the input and output. The input is -30dBm, and the output is -10dBm.

© 1994-2020
All Rights Reserved.

Harmonic Balance

There are three ways to run Harmonic Balance simulations in SpectreRF. These are:

- pss-hb
- qpss-hb
- hb

Qpss with the harmonic balance engine selected and hb are the exact same analysis, however the hb *Choosing Analyses* form has all the latest improvements, and the qpss form does not. If you are considering using qpss-harmonic balance, it is strongly recommended that you use the hb *Choosing Analyses* form instead.

Harmonic balance in the pss analysis is slightly different than in hb and qpss harmonic balance. In pss, it is always a single-tone simulation. This means that all the harmonics of the beat frequency are calculated by pss instead of just the harmonics of the signals and the mixing products.

Also, there are some analysis options that are available directly in the ADE Explorer GUI for pss-harmonic balance that are not in the ADE Explorer GUI for hb. Although you do not require these options, you can still use an option that is not in the ADE Explorer GUI for hb by typing `option_name=value` in the *AdditionalParams* field for the option that you want to use in hb.

© 1994-2020 All Rights Reserved.

In the example below, there are two input frequencies at 2.4GHz and 2.5GHz.

© 1994-2020
All Rights Reserved.

The simulation was run with 135 harmonics so that the fifth harmonic of 2.5GHz was captured. Usually, running hb or qpss-harmonic balance is faster because fewer harmonics need to be calculated. This example is for reference only.

© 1994-2020 All Rights Reserved.

© 1994-2020
All Rights Reserved.

The output spectrum shows that all the harmonics of 100MHz are calculated.

Again note that hb or qpss-harmonic balance requires fewer harmonics to be calculated because just the significant harmonics that the circuit actually produces are calculated.

Other than this difference, pss-harmonic balance is exactly the same as hb. For more information on harmonic balance, refer to Chapter 3, “Frequency Domain Analyses: Harmonic Balance”.

© 1994-2020 All Rights Reserved.

DC Principles

In order to understand the pss analysis, the fundamentals of simulation need to be discussed. Since the transient starts from a DC analysis that provides the time-zero timepoint, the DC Principles are reviewed. Since shooting pss is an extension to the transient analysis, Transient Principles on page 664 are also provided as a review.

In any simulation, there has to be an equal number of simultaneous equations as unknowns in the circuit. In simulation, the KCL equations at each node are used. The fundamental assumption is that the sum of the currents at each node has to be zero.

The capacitors and inductors can be removed from the circuit because they have no effect on the DC solution. The resistors have linear current and voltage relationships and go directly into a linear algebra matrix. Sources also go directly into the matrix. These values do not change from iteration to iteration. The nonlinearities do not have constant resistance with respect to the applied input voltage.Therefore, they are replaced by resistors and current sources which go into the matrix. Now all the entries in the matrix are linear, and the matrix can be solved. An iterative technique is used to calculate the DC bias point.

Consider the circuit shown below.

© 1994-2020
All Rights Reserved.

On a current versus voltage plot, the system looks, as shown below.

The intercept where the battery/resistor current coming in is equal to the diode current going out is the exact solution to the problem. Since computers cannot solve algebraic equations, an iterative approach is used to calculate the solution.

A starting voltage is assumed for each nonlinearity in the circuit. The default is to assume that the device is on. For a diode, that means about 0.7 volts across the device.

In the device model, the I versus V curve is available, and the ability to calculate the tangent at any voltage is also provided. For the first iteration, a line is drawn vertically from the starting point, and at the point where the vertical line intersects the I-V curve, the tangent is calculated. The nonlinear junction is replaced with a resistor and a current source. The current source value is the negative Y-Axis intercept of the tangent, and the conductance of the resistor is the slope of the tangent. The reason for doing this is that the resistor and current sources go directly into a linear algebra matrix that can be inverted and solved for voltages

© 1994-2020 All Rights Reserved.

on each node. From that solution, a new tangent is calculated, that value is placed in the matrix, and it is solved again. This is shown below for the first two iterations.

Note that the iterations can continue forever, always getting closer to the correct answer, but never achieving the correct answer.

This process is used for every nonlinearity in the circuit. Note that for a more complicated device, there may also be nonlinear-controlled sources to model the output current.

Convergence

The iterations stop (and the circuit has converged) when the answer is close enough. Close enough is defined by the settings of the convergence options `reltol`, `vabstol`, and `iabstol`. Just after the linear solution is calculated, the current from the linear solution is compared with the nonlinear current from the nonlinear equation for all the nonlinearities in

© 1994-2020 All Rights Reserved.

the circuit. When the difference in current is less than `reltol` * nonlinear current + `iabstol` for all the devices, a second check is made. When the change in voltage from the last iteration to this one for all nodes in the circuit is less than `reltol` * (the larger of the last or this iteration voltage) + `vabstol`, then a KCL check is made for all the nodes. When the sum of the currents at each node is less than `reltol` * (largest current into the node) + `iabstol`, the solution has converged. All three criteria need to be met everywhere in the circuit to have the iterations stop. When this happens, the circuit converges and the iterations stop. Again, note that the solution is not exact. To get a more accurate solution, `reltol` and `vabstol` need to be reduced from the defaults. More accuracy in DC usually costs a few more iterations.

The default for `reltol` is `1e-3` (0.1%). The default for `vabstol` is `1e-6` (1 microvolt). The default for `iabstol` is `1e-12` (1 picoamp). By default, when the answer is within 0.1% plus a little bit, the simulator stops iterating.

With the default values, note that when the solution is 1 millivolt, the error that is allowed is 0.1% of 1 millivolt + 1 microvolt or 2 microvolts. When the solution is larger than 1 millivolt, the relative term dominates the error tolerance. Similarly, when the current reaches 1 nanoamp, the relative tolerance becomes dominant. Because most circuits have voltages and currents larger than these, the relative term (`reltol`) usually dominates the error tolerances. The absolute terms (`vabstol` and `iabstol`) apply just to a very small region near zero.

The voltage that exists from the sources in the circuit at time zero is applied, and the solution of the DC algorithm is used as the first timepoint.

Skipdc

The default for `skipdc` is `no`. Normally, a DC solution is run for the initial transient solution at time zero.

In some cases, the time zero timepoint does not converge. Using the `skipdc` option allows the simulation to proceed to the transient without a DC solution for the time zero point. In this case, an assumed solution is used as the starting point. For the nodes that have batteries connected to ground, the assumed solution is the battery voltage. For nodes that have initial conditions specified, the initial condition is used. For all other nodes, zero volts is assumed.

When `skipdc` is set to `yes`, the assumed solution is used at time zero and the transient sources start immediately. Sometimes, this causes problems for the transient, therefore, setting `skipdc` to `sigrampup` causes the time-varying to start at minus 10 percent of the stop time in the `tstab` and ramp up from zero to the final value at zero seconds. When the default for `skipdc` causes problems for the transient, easing the input signal can help the transient get started.

© 1994-2020 All Rights Reserved.

Transient Principles

In the transient, the input voltage is allowed to vary as a function of time, and the capacitor and inductors are added to the circuit. The capacitors are modeled using a resistor and a current source in parallel, and the inductors are modeled as a battery in series with a resistor. The values of the resistors, batteries, and current sources are adjusted at each timepoint to accurately model the correct behavior of the capacitor and inductor. Note that the matrix is still scalar (No real and imaginary part). Keeping the matrix scalar speeds up the solution at each timepoint.

The system is broken into individual timepoints, and a solution is made at each timepoint.

Convergence

At each timepoint, the system is solved for using an iterative process, just like the DC solution. The difference is that in the transient, history is used to predict the movement of the circuit, so that the number of iterations at each timepoint is reduced. The same convergence criteria is used in the transient as in the DC analysis with the exception of where the `reltol` * `voltage` or `reltol` * `current` terms come from in the circuit. This is controlled by the `relref` option.

Relref

Consider a sine wave, as shown below.

© 1994-2020 All Rights Reserved.

- Pointlocal criteria is `reltol` * the voltage or current at this node at this iteration or the last iteration, whichever is larger. Note that as convergence is attained, the solution is almost exactly the sinusoid, so when the voltage is small, the uncertainty is small. When the voltage is large, the uncertainty is large. This is shown below.

  Although this is the most accurate solution, more time is required for the simulation because of the very small error tolerances near zero. Usually, the local criteria, which is shown next is preferable.

- Local criteria is `reltol` * the voltage or `reltol` * `current` at this node at this or the previous iteration, or at any solution in the past. With a sinusoid starting at zero, the uncertainty is zero and gets larger as the voltage gets larger. Once the voltage reaches

© 1994-2020 All Rights Reserved.

the peak, the uncertainty is constant after that. This is because `reltol` * peak voltage is always larger than `reltol`* the current solution.

The error tolerance is larger after the first peak compared to `pointlocal`, but it is never larger than the value at the peak. This is a good compromise for accuracy of the simulation.

- Global criteria is `reltol` * largest voltage anywhere in the circuit now or at any time in the past. For most circuits, the largest voltage in the circuit is the power supply voltage and the largest current is the power supply current. Since the power supply voltage and

© 1994-2020 All Rights Reserved.

current is usually larger than the node voltages and currents, the error is larger, as shown below.

Note that when global criteria are used, the error tolerance gets much larger for low voltage nodes and low current branches. Because of this, the accuracy degrades when using global criteria.

- Note that for `pointlocal` and for local criteria at `time = zero`, when the voltage is zero, `reltol` * voltage is also zero. Since every iteration produces a different solution, an absolute term needs to be added to the relative term so that convergence can be obtained when the voltage or current signal goes through zero. For voltages, this is set by the option `vabstol`, which defaults to 1 microvolt. (`1e-6`) For currents, this is set by

© 1994-2020 All Rights Reserved.

the option `iabstol` which defaults to 1 picoamp (`1e-12`). This small absolute tolerance is added to the relative term at all timepoints. The absolute term is shown below.

The `relref` option affects the transient-based analysis only (tran, pss shooting, qpss shooting, and envelope shooting). It uses different combinations of criteria (`pointlocal`, `local`, `global`) for the different settings of `relref`.

	Pointlocal	**Alllocal**	**Sigglobal**	**Allglobal**
Device Current Check	Pointlocal	Local	Global	Global
Delta-V Check	Pointlocal	Local	Global	Global
KCL Check	Pointlocal	Local	Local	Global

Note that in the order of most accurate to least accurate, the `relref` settings are `pointlocal`, `alllocal`, `sigglobal`, and `allglobal`. The default for most SpectreRF simulations is `alllocal`.

© 1994-2020 All Rights Reserved.

Integration Methods

Trapezoidal Method

In the transient, numerical integration is used for integrating current to get voltage on capacitors and for integrating voltage to get current in inductors. One method is to assume that the current is constant at the average current that flows during the interval. Because the current is constant, the voltage is a ramp during the timestep. The waveform that might be produced is shown below.

Note that the real waveform is usually a curve. This is shown below.

© 1994-2020 All Rights Reserved.

Note that the area between the straight line and the curve is the charge that is in the physical system, but not in the simulation. Errors are introduced in the simulation because of a finite timestep that is in the simulation. The timestep cannot go infinitely small because it would take infinitely long to run the simulation. As a consequence, numerical integration errors are inherent in the time-domain simulation because of the finite timestep.

This error can be controlled by making the timestep smaller, as shown below.

Note that by taking twice as many timepoints, the sum of the errors is much less than the previous case. One way to accomplish this is to set a small maximum timestep. Although this produces the most accurate waveform, the runtime cost is usually quite high because the maximum timestep is used everywhere in the simulation.

When the waveform is at a constant value, there is no numerical integration error. The place where the error occurs is the curvature that is produced by the circuit. Spectre controls the timestep based on the curvature that is produced by the circuit. Because of this, the timestep

© 1994-2020 All Rights Reserved.

is inherently a variable, as shown in the below waveform. The X symbols on the waveform indicate the place where a timepoint exists in the transient output data.

Note that where the waveform is flat, the timestep bumps out, and where there is a high curvature, lots of points are taken. In this way, the accuracy of the waveform is preserved with the least possible number of timepoints. This is done to reduce the runtime and is the normal method of timestep control.

The trapezoidal method is generally good for normal transient simulations because the method does not exaggerate or damp any ringing that the circuit might produce. Although it is not visible in the waveform above, the trapezoidal method has ringing. At each point, there is an error term that alternates its sign at each timepoint. First a bit up, then a bit down.

Gear2 method

In the Gear2 method, the current is assumed to be a polynomial fit in the timestep. This requires the simulation history to determine the curve. Because of this, the simulation takes

© 1994-2020 All Rights Reserved.

about 5% longer to run compared to trapezoidal. Because the current is more accurately represented, the method is a bit more accurate than trapezoidal. It also slightly numerically damps any ringing that the circuit might produce. This effect disappears completely when the timesteps are small, which is usually accomplished by setting `reltol` to `1e-4` or smaller. For very high Q circuits, small numerical integration errors cause large errors in the voltage in the resonator. For that reason, harmonic balance is usually preferable to shooting for high Q circuits. If you must simulate a high Q circuit using the transient analysis, always use the `traponly` numerical integration method.

Controlling Accuracy

The simulator works by setting a limit for the curvature. This limit is controllable by setting `lteratio` and/or `reltol`. The product of `reltol` times `lteratio` sets the actual limit. Increasing `lteratio` sets a higher amount of curvature and leads to more errors when evaluating the numerical integration. The output with `lteratio` set to 10 times the default, is shown below.

© 1994-2020
All Rights Reserved.

The waveshape is substantially similar. In this case, it is OK to allow more numerical integration errors. If the details of a small portion of the waveform need to be accurate, zoom in to the time scale that is needed, as shown below.

© 1994-2020
All Rights Reserved.

Note the obvious piecewise linear behavior in the waveform. This is an indicator that at this time scale, the error is unacceptable. `Lteratio` and `reltol` were reduced for the following waveform.

The waveshape is substantially changed. This is because the numerical errors were significant at the time/voltage scale that we needed for the measurement.

There is still a bit of piecewise linear behavior. Reducing `reltol` and `lteratio` further will only be slightly more accurate and the runtime will be much longer. In simulation, a trade-off of accuracy versus runtime needs to be made using `reltol`, `vabstol`, and `lteratio` to get the accuracy that is needed without excessive runtimes.

© 1994-2020
All Rights Reserved.

Shooting PSS

Overview

Shooting is a way of calculating the steady-state waveforms of a periodic system. Any waveshape is allowed as an input with the limitation that the waveform must be periodic. First, a transient is run for the duration set by the *Stop time* (`tstab`) option. Next, one period of the pss beat frequency is simulated while still using the normal transient algorithm. Then, the transient continues for one period of the waveform with the addition of saving all the solution matrices at each timestep. This is called the shooting window.

At the end of simulating this period of the input, the beginning and ending states are compared. If all the voltages and currents are equal, the steady-state solution has been achieved, and the simulation stops. In general, because the transient analysis starts from a settled DC solution at time zero, there is still settling behavior going on at the end of the first time through the shooting window. This causes the solutions to be different at the beginning and ending of the shooting window. The simulator now looks through all the solution matrices which contain information about the behavior of the circuit and the nonlinear resistors, capacitors, and inductors in the circuit, Using this information, an estimate is made of the final settled voltage for every node in the circuit. This is brought back as the starting point for the shooting window, and one period is run again. At the end, the voltages and currents are again compared to see if they are equal. This process continues until the beginning and ending states are equal. This process is called the shooting Newton method.

© 1994-2020 All Rights Reserved.

Graphically, this is shown below.

The estimate for every node is brought back as the starting point, and the period just after 15 nsec is simulated again. This continues until the beginning and ending voltages and currents almost match.

Note that PSS Shooting is really an extension to the transient analysis. All the concepts and options that were discussed before in the transient review section apply to shooting pss as well.

Oscillators

The algorithm is slightly different for oscillators. First, the tstab interval is simulated. Next, a single period of the estimated frequency from the *Choosing Analyses* form is simulated. So far, this is exactly the same as driven pss. To achieve reliable convergence, the oscillator must reach near the steady-state behavior during this phase.

Because the oscillator may also have a frequency divider at the output, additional four periods of the estimated frequency in the pss *Choosing Analyses* form are simulated. During this time, Spectre looks for a frequency divided output.

© 1994-2020 All Rights Reserved.

Next, one period of the lowest frequency in the circuit is determined. If the oscillator does not have a frequency divider, then one period of the oscillator output is solved for. If the oscillator has a divider, one period of the divided down frequency is solved for.

Once one period is determined, then the steady-state waveforms for all the nodes and the period are solved for so that the beginning and ending voltages and currents almost match, and the period is stable for at least two iterations. Because the period is an additional variable to be solved for compared to a driven circuit where the period is known, oscillators typically take more iterations to converge to the solution.

Oscillator Tuning Mode

Pss has the ability to tune an oscillator to a specified frequency, and then run the small-signal analyses (usually noise) that are specified. This is typically used in sweeps and in Monte Carlo simulations where the phase noise is desired at a specified frequency.

Tuning mode adjusts a parameter in the circuit to produce the set frequency target. This is done automatically when tuning mode is enabled without setting any sweep parameters or interpolation of the resulting curves. When the tuning frequency is reached, any small-signal analyses like noise are run. This allows the simulator to tune the oscillator to a specified frequency and then make a noise measurement. This is useful in Monte Carlo analysis to see how the oscillator performs with process variations. The parameter to be tuned can be a variable, temperature, or a specified device parameter. Oscillator tuning mode is also supported in the hb analysis. See Chapter 3, “Frequency Domain Analyses: Harmonic Balance,” for details.

In this mode, the target frequency to tune to is the frequency that is specified in the pss *Choosing Analyses* form as the fundamental frequency. You specify a parameter that is to be varied to achieve the desired frequency. This can be a variable, a device parameter, or temperature. When the analysis runs, the oscillator will be tuned to the desired frequency, and then all the small-signal analyses will be run.

Direct plot functions have been added to plot the tuning parameter.

Delay Time

All the signal sources have a property called delay time. During the delay time, the source stays at a constant value, and then the waveform starts. The system does not become periodic until the longest delay time has elapsed. Delay times longer than one period of the output waveform of a source should be avoided. The beginning of the tstab interval is the longest delay time in the sources of the circuit.

© 1994-2020 All Rights Reserved.

Piecewise Linear Sources for Power Supplies

In some cases, nonperiodic piecewise linear sources are used. This is one way to start an oscillator by giving the power supply a small bump just after the transient analysis starts. In this case, set the first point at zero seconds and 80% to 90% of the full supply voltage, and specify a second point about half period of the operating frequency at the full supply voltage. Do not specify a point at one second (or some other fairly large time) and the supply voltage. Spectre will maintain the voltage of a source if the simulation time exceeds the last point from a piecewise linear source. If there is a one second point, then something that is usually at a high frequency has to simulate for one second before the circuit becomes periodic. Tstab would start at the last point in the piecewise linear waveform.

Tstab

In the pss Choosing Analyses form, there is a property called *Stop time (tstab)*. The default value for tstab is zero. However, if a value is specified for the parameter, then a normal transient is run to a stop time equal to tstab. For driven circuits, it is seldom necessary to set tstab, but oscillators usually require it to be set.

For shooting pss, when *Run transient?* is set to *yes*, the *Detect Steady State* option is enabled automatically. This option can be selected to switch from running tstab directly in the shooting window when steady state is detected, if this occurs before the stop time specified in the *Stop Time(tstab)* field. When you set the *Run transient?* option to *Decide automatically,* tstab is run until steady state is achieved and then switches to the shooting phase automatically.

Dynamic parameters are supported in tstab just like in the transient analysis. Typically only reltol is changed over the tstab interval with loose (large) values of reltol at the beginning of tstab and then tighter values toward the end. A single option is supported in the GUI. Note that the default values are already fairly loose in the tstab interval by default.

Beat Frequency

The beat frequency is the periodicity of the system. If a single input frequency is applied, it is the frequency of the input signal. For frequencies that have inconvenient frequencies, but easy periods, a selection is available to set the period instead of the frequency.

If there are two input frequencies, the highest common multiple frequency must be set as the beat frequency. The limitation is that an integer number of cycles must be simulated for all the input frequencies. This is required because pss checks the differences between the currents and voltages at the beginning and end of the shooting window and this difference must be zero. This forces an integer number of cycles for all the sources in the circuit to be simulated.

© 1994-2020 All Rights Reserved.

As an example, if a 1 GHz frequency is applied to the system, the beat frequency (periodicity) of the system is 1 GHz. If 1 GHz and 1.1 GHz frequencies are applied, the beat frequency is 100 MHz. In this system, 10 cycles of 1GHz and 11 cycles of 1.1 GHz need to be simulated to get the steady-state waveform. If 1 GHz and 1.001GHz need to be simulated, the beat frequency is 1 MHz. This is not a practical pss simulation because 1000 cycles of 1 GHz and 1001 cycles of 1.001 GHz need to be simulated, and at least 3003 harmonics need to be calculated to measure the third harmonic of 1.001 GHz. The time required for the simulation and the memory requirement is excessive for this case. In this case, using qpss or hb is much faster and requires much less memory.

For oscillators, an estimate of the oscillator frequency is required in the *Beat Frequency* field. If the linear oscillator initial condition is used, the frequency range needs to be from 0.5 to 1.5 times the actual oscillation frequency. If the default oscillator initial condition is used, then the frequency must be between 0.25 and four times the actual frequency of oscillation.

Number of Harmonics

In the case of PSS Shooting, this is a post-processing specification that might have an effect on the minimum number of timepoints in the solution. This parameter specifies the number of harmonics to calculate in the Fourier transform that is run after the time-domain simulation to calculate the harmonics of the large-signal response. The Fourier transform is not an FFT that requires exactly spaced samples of the waveform. Rather, it is Fourier integral-based, which uses the timepoints that were actually calculated by the pss analysis as the input. Since the waveform that the Fourier-integral based algorithm sees are the actual timepoints, all the harmonics specified are accurate, even if you only specify one or two harmonics. Zero (0) is a perfectly legal value for the number of harmonics, and it just means that no Fourier transform will be run after the time-domain solution is calculated.

If the number of harmonics is raised above 10, then more timepoints will be forced in the pss solution in order to keep the accuracy of the frequency-domain results very high. 20 timepoints will be run in the period of the highest harmonic. Therefore, if maximum harmonics is set to 20, a minimum of 400 timepoints are forced in the pss analysis. If a large number of harmonics is specified, runtime and memory requirements get large. Unless you really need a lot of harmonics, do not set a large number.

Accuracy Defaults (errpreset)

Errpreset sets the default values for six options in order to tune the simulator for your needs. Use `conservative` if you need high accuracy. `Moderate` is a general-purpose setting, and should be used for most simulations. `Liberal` is not recommended.

© 1994-2020 All Rights Reserved.

The following chart summarizes the options and the values for driven circuits.

errpreset	Reltol[1]	Relref	Method	Lteratio[2]	Steadyratio	Maxstep
Liberal	1e-3	Sigglobal	Traponly	3.5	0.001	Period/50
Moderate	1e-3	Alllocal	Gear2only	3.5	0.001	Period/200
Conservative	1e-4	Alllocal	Gear2only	10 or 3.5	0.01	Period/200

1. Reltol: This value is used as a maximum value in the shooting window. The value used in the tstab interval is set by the option reltol.
2. Lteratio: For conservative, 10 is used in the shooting window unless the option reltol is set to 1e-4*10/3.5 or smaller. If reltol is set to 1e-4*10/3.5 or smaller, lteratio defaults to 3.5. The value used for lteratio in the tstab interval is the same as that used in the shooting window.

For example, if `errpreset` is set to `moderate` and `reltol` is set to `3e-4`, `reltol` during the tstab interval is `3e-4`, and `reltol` in the shooting window is also `3e-4` because it is smaller than `1e-3`, which is the maximum value that can be used in the shooting window. `Lteratio` is 3.5 for the tstab interval and for the shooting window.

If `errpreset` is `conservative` and `reltol` is set to `3e-4`, then `reltol` in the tstab interval is `3e-4`, and in the shooting window it is `1e-4`. This is because the maximum value for `reltol` in the shooting window for `conservative` is `1e-4`. `Lteratio` is `10` for both the tstab interval and the shooting window.

For oscillators, the defaults are slightly different.

errpreset	Reltol[1]	Relref	Method	Lteratio[2]	Steadyratio	Maxstep
Liberal	1e-3	Sigglobal	Traponly	3.5	0.001	Period/50
Moderate	1e-4	Alllocal	Gear2only	3.5	0.001	Period/200
Conservative	1e-5	Alllocal	Gear2only	10	0.1	Period/200

1. Reltol: This value is used as a maximum value in the shooting window. The value used in the tstab interval is set by the option reltol.
2. Lteratio: For conservative, 10 is used in the shooting window unless the option reltol is set to 2.85e-4 or smaller. If 2.85e-4 or smaller is used, lteratio defaults to 3.5. The value used in the tstab interval is the same as that used in the shooting window.

© 1994-2020 All Rights Reserved.

PSS Shooting Convergence

As discussed earlier, the waveform is iteratively solved for during the shooting window. This means that exact solutions are never obtained because there is another iteration that will produce a closer answer. The iterations must stop when the solution is accurate enough. This is set using the convergence options `reltol`, `vabstol`, `iabstol`, `lteratio`, `errpreset`, and an option called `steadyratio`. The iterations stop when all the voltages in all the nodes and all the currents in all the branches match at the beginning and end within the following tolerance:

```
(Convergence Criteria) * Lteratio * Steadyratio
```

The convergence criteria depend on the setting of `reltol`, `vabstol`, `iabstol`, `errpreset`, and the setting of `relref`. `Lteratio` defaults to `3.5` or `10`, depending on the setting of `reltol` and `errpreset`.

Note: If the PSS analysis does not converge, the unconverged solution is not saved.

`Steadyratio` depends on the setting of `errpreset`. For more details regarding this option see Steadyratio (Shooting Only) on page 713.

© 1994-2020 All Rights Reserved.

ADE Implementation and Numerical Noise Floor

Next, we will discuss how SpectreRF is implemented into ADE Explorer and also provide information about how the simulation accuracy is affected by setting a number of simulator options.

A transistor-level LNA is shown below.

© 1994-2020
All Rights Reserved.

The input frequencies for the transistor-level LNA are set to 2.4 GHz and 2.5 GHz, as shown in the ADE Explorer window below.

Note that this setup is approaching the limit of what pss can do because 25 periods of the higher input frequency need to be simulated in the shooting window. If the frequencies were closer together, qpss or hb would be much more efficient.

© 1994-2020 All Rights Reserved.

The pss *Choosing Analyses* form is shown below.

© 1994-2020
All Rights Reserved.

All the sources in the circuit are read in to the *Fundamental Tones* table at the top of the *Choosing Analyses* form. When the *Auto Calculate* option is selected, ADE Explorer calculates the periodicity of the circuit and displays it in the grayed-out *Beat Frequency* field.

Note: When *Auto Calculate* is selected, manual editing of the *Beat Frequency* field is not allowed.

50 harmonics have been set to allow the second harmonic of the 2.5GHz tone to be calculated.

Driven circuits usually do not require tstab to be set. For circuits that do have tstab set simulation options can be changed at different times during the tstab interval by using dynamic parameters. This example shows reltol starting very loose, and then switching to a

© 1994-2020 All Rights Reserved.

more accurate value at 10nsec. The setting of reltol during tstab is shown in the spectre output window.

```
/servers/scratch03/kamal/spectrerf131/testcase131/docs_dat
File  Edit  View  Help                                    cadence

    temp = 27 C
    tnom = 27 C
    tempeffects = all
    method = traponly
    lteratio = 3.5
    relref = sigglobal
    cmin = 0 F
    gmin = 1 pS
    rabsshort = 1 mOhm


Notice from spectre at time = 10 ns during periodic steady state analysis `pss'
    The value of parameter 'reltol' was set to 0.0001 at time 1e-08 s.

    pss: time = 12.76 ns     (2.5 %), step = 21.91 ps     (4.3 m%)
    pss: time = 38.26 ns     (7.5 %), step = 26.25 ps    (5.15 m%)
    pss: time = 63.76 ns    (12.5 %), step = 27.74 ps    (5.44 m%)
    pss: time = 89.25 ns    (17.5 %), step = 27.57 ps    (5.41 m%)
    pss: time = 114.8 ns    (22.5 %), step = 33.33 ps    (6.54 m%)
    pss: time = 140.3 ns    (27.5 %), step = 29.67 ps    (5.82 m%)
    pss: time = 165.8 ns    (32.5 %), step = 33.33 ps    (6.54 m%)
    pss: time = 191.3 ns    (37.5 %), step = 29.43 ps    (5.77 m%)
    pss: time = 216.8 ns    (42.5 %), step = 33.33 ps    (6.54 m%)
Steady-state is detected after 240.008 ns.

Notice from spectre during periodic steady state analysis `pss'.
    The value of parameter 'reltol' was set to 0.0001 at time 2.40008e-07 s.

Total time required for tstab analysis `pss': CPU = 3.69744 s, elapsed = 3.9807
Time accumulated: CPU = 3.69744 s, elapsed = 3.98082 s.
Peak resident memory used = 40.3 Mbytes.

Conv norm = 3.41e+03, max dI(I0.I28.I10.ls1:1) = 585.221 uA, took 3.39 s.

Important parameter values in pss iteration:
    start = 230.008 ns
    outputstart = 0 s
    stop = 240.008 ns
    period = 10 ns
    maxperiods = 20
    steadyratio = 10e-03
    step = 510 ns
90  >                                                    L1      C1
```


© 1994-2020
All Rights Reserved.

Accuracy and Settings and Trade-Offs

A comparison is made for several settings. Moderate and conservative settings are used.

After that, values for *reltol* and *vabstol* are specified, as shown below in the *Simulator Options* form. This form can be opened by selecting *Simulation* - *Options* - *Analog* in ADE Explorer.

First, `reltol=1e-5` and `vabstol=3e-8` are used, and then `reltol=1e-6` and `vabstol=1e-8` are used. Finally, a lot of timepoints are forced by setting the maximum harmonics to 1000. For this test, `reltol` and `vabstol` are kept at `1e-6` and `1e-8`.

© 1994-2020
All Rights Reserved.

The spectral result from all the simulations is shown below.

A marker is placed at 1.1 GHz in order to measure the numerical noise floor of the simulation.

The top is with `errpreset` set to `moderate`.

The second is `conservative`.

The third is `reltol=1e-5` and `vabstol=3e-8`.

The fourth is `reltol=1e-6` and `vabstol=1e-8`.

The final one is with 1000 harmonics set. The display is scaled to show only the first 30 harmonics.

Note that the numerical noise floor drops with the steps that are taken to increase accuracy. The runtime and memory requirement also increases for each successive level of accuracy.

© 1994-2020
All Rights Reserved.

This is a good general series of steps to improve the accuracy of a shooting simulation. The basic process is to set up and run the simulation, make the accuracy better, and run the simulation again. If the answer does not change, then the original setting is good enough. If the answer changes, increase the accuracy again. Continue until you find the setting of the options that cause the answer to be accurate without excessive runtime.

Note that there is a practical minimum numerical noise floor at about -135dBm. Although it might be possible to lower the noise floor a bit more by taking even more timepoints in the simulation, the time and memory requirements usually are not worth the lower noise floor. Still, -142dBm is a pretty good numerical noise floor of about 3e-8 volts RMS.

Additionally, you might try using the *highorder* option to improve accuracy. This is shown in the pss options section. This option requires significantly longer to run but produces a more accurate result.

Which Engine Should be Selected?

Conceptually, shooting and harmonic balance produce the same answer as long as enough harmonics and/or a high enough oversample factor are used in hb and the accuracy of shooting is controlled well enough for your requirement. In general, if the accurate waveform is desired for a system with square waves in the solution, then shooting should be selected. If the system is very nonlinear like a sample-and-hold or a switched-capacitor filter, shooting should always be used. If the waveforms are nearly sinusoidal, then harmonic balance should be selected. If it is something in between, then try both engines and use the one that produces the fastest runtime. Note that even for square wave systems, harmonic balance can be used effectively as long as the time-domain waveform is not desired, and oversample factor is set to at least four. For more details on harmonic balance, see Chapter 3, “Frequency Domain Analyses: Harmonic Balance”.

© 1994-2020 All Rights Reserved.

Important Outputs in the Spectre.out File

```
/servers/scratch03/kamal/spectrerf131/testca
File  Edit  View  Help                                   cadence

Circuit inventory:
                 nodes 250
                 balun 2
               bsim3v3 68
             capacitor 58
              inductor 27
      mutual_inductor 12
               phy_res 30
                  port 2
              resistor 162
               vsource 3

Analysis and control statement inventory:
                  info 6
                   pss 1

Output statements:
                .probe 0
              .measure 0
                  save 0


Notice from spectre during initial setup.
    APS Enabled.
    5 warnings suppressed.

Time for parsing: CPU = 4 ms, elapsed = 9.50217 ms.
Time accumulated: CPU = 722.889 ms, elapsed = 5.81927 s.
Peak resident memory used = 39.8 Mbytes.

~~~~~~~~~~~~~~~~~~~~~~~~~~
Pre-Simulation Summary
~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~

Warning from spectre.
    WARNING (SPECTRE-16707): Only tran supports psfxl format,
Notice from spectre.
    Multithreading is disabled due to the size of the design


*****************************************************
Periodic Steady-State Analysis `pss': fund = 100 MHz
*****************************************************
21  >                                                L1    C1
```


© 1994-2020
All Rights Reserved.

The circuit inventory (highlighted in red above) lists the contents of the circuit.

If APS is used, it is noted in the section highlighted in green above.

The section highlighted in blue below contains the option settings for tstab simulation.

© 1994-2020 All Rights Reserved.

The section highlighted in brown below lists the options for shooting iterations.

```
/servers/scratch03/kamal/spectrerf131/testcas
File  Edit  View  Help                                   cadence

Important parameter values in tstab integration:
    start = 0 s
    outputstart = 0 s
    stop = 510 ns
    period = 10 ns
    maxperiods = 20
    step = 510 ps
    maxstep = 400 ps
    ic = all
    useprevic = no
    skipdc = no
    reltol = 1e-03
    abstol(V) = 1 uV
    abstol(I) = 1 pA
    temp = 27 C
    tnom = 27 C
    tempeffects = all
    method = traponly
    lteratio = 3.5
    relref = sigglobal
    cmin = 0 F
    gmin = 1 pS
    rabsshort = 1 mOhm

    pss: time = 12.77 ns     (2.5 %), step = 33.33 ps    (6.54
    pss: time = 38.27 ns     (7.5 %), step = 33.33 ps    (6.54
Steady-state is detected after 40 ns.
Total time required for tstab analysis `pss': CPU = 1.45978 s,
Time accumulated: CPU = 1.45978 s, elapsed = 6.57676 s.
Peak resident memory used = 41.8 Mbytes.

Conv norm = 3.9e+03, max dI(I0.I28.I10.1s1:1) = 670.269 uA, tool

Important parameter values in pss iteration:
    start = 30 ns
    outputstart = 0 s
    stop = 40 ns
    period = 10 ns
    maxperiods = 20
    steadyratio = 1e-03
    step = 510 ps
    maxstep = 50 ps
    ic = all
    useprevic = no
    skipdc = no
    reltol = 1e-03
    abstol(V) = 1 uV
    abstol(I) = 1 pA
    temp = 27 C
    tnom = 27 C
    tempeffects = all
    errpreset = moderate
    method = gear2only
    lteratio = 3.5
```


© 1994-2020 All Rights Reserved.

When the `Conv_Norm` number gets less than one, the iterations stop. In this case, the error is 6.03e-3 times the maximum allowed, and the largest mismatch is I0.I28.I6.Is1 terminal 1, and the mismatch is 10.1339 microamps (highlighted in green below).

```
/servers/scratch03/kamal/spectrerf131/testcase131/docs_
File  Edit  View  Help                                          cadence
    pss: time = 44.75 ns    (47.5 %), step = 500 fs         (5 m%)
    pss: time = 45.25 ns    (52.5 %), step = 500 fs         (5 m%)
    pss: time = 45.75 ns    (57.5 %), step = 500 fs         (5 m%)
    pss: time = 46.25 ns    (62.5 %), step = 500 fs         (5 m%)
    pss: time = 46.75 ns    (67.5 %), step = 500 fs         (5 m%)
    pss: time = 47.25 ns    (72.5 %), step = 500 fs         (5 m%)
    pss: time = 47.75 ns    (77.5 %), step = 500 fs         (5 m%)
    pss: time = 48.25 ns    (82.5 %), step = 500 fs         (5 m%)
    pss: time = 48.75 ns    (87.5 %), step = 500 fs         (5 m%)
    pss: time = 49.25 ns    (92.5 %), step = 500 fs         (5 m%)
    pss: time = 49.75 ns    (97.5 %), step = 500 fs         (5 m%)
Conv norm = 6.03e+03, max dI(I0.I28.I6.Is1:1) = 10.1339 uA, took 9.25 s.

========================================
`pss': time = (40.0019 ns -> 50.0019 ns)
========================================
    pss: time = 40.25 ns     (2.5 %), step = 500 fs         (5 m%)
    pss: time = 40.75 ns     (7.5 %), step = 500 fs         (5 m%)
    pss: time = 41.25 ns    (12.5 %), step = 500 fs         (5 m%)
    pss: time = 41.75 ns    (17.5 %), step = 500 fs         (5 m%)
    pss: time = 42.25 ns    (22.5 %), step = 500 fs         (5 m%)
    pss: time = 42.75 ns    (27.5 %), step = 500 fs         (5 m%)
    pss: time = 43.25 ns    (32.5 %), step = 500 fs         (5 m%)
    pss: time = 43.75 ns    (37.5 %), step = 500 fs         (5 m%)
    pss: time = 44.25 ns    (42.5 %), step = 500 fs         (5 m%)
    pss: time = 44.75 ns    (47.5 %), step = 500 fs         (5 m%)
    pss: time = 45.25 ns    (52.5 %), step = 500 fs         (5 m%)
    pss: time = 45.75 ns    (57.5 %), step = 500 fs         (5 m%)
    pss: time = 46.25 ns    (62.5 %), step = 500 fs         (5 m%)
    pss: time = 46.75 ns    (67.5 %), step = 500 fs         (5 m%)
    pss: time = 47.25 ns    (72.5 %), step = 500 fs         (5 m%)
    pss: time = 47.75 ns    (77.5 %), step = 500 fs         (5 m%)
    pss: time = 48.25 ns    (82.5 %), step = 500 fs         (5 m%)
    pss: time = 48.75 ns    (87.5 %), step = 500 fs         (5 m%)
    pss: time = 49.25 ns    (92.5 %), step = 500 fs         (5 m%)
    pss: time = 49.75 ns    (97.5 %), step = 500 fs         (5 m%)
Conv norm = 11.8e-03, max dI(I0.I28.I6.Is1:1) = 19.879 pA, took 9.1 s.

Opening the PSF file ../psf/pss.td.pss ...

Opening the PSF file ../psf/pss.fd.pss ...
Number of accepted pss steps =              50193
Initial condition solution time: CPU = 83.987 ms, elapsed = 84.5559 ms.
pss: The steady-state solution was achieved in 3 iterations.
Number of accepted pss steps =              20000
Total time required for pss analysis `pss': CPU = 37.7153 s, elapsed = 25.65
Time accumulated: CPU = 38.2572 s, elapsed = 26.7515 s.
Peak resident memory used = 458 Mbytes.

modelParameter: writing model parameter values to rawfile.

25  >                                                   L323    C60
```


© 1994-2020
All Rights Reserved.

When the Conv_Norm is less than one, the iterations stop (highlighted in red above).

The total number of timesteps in the solution is listed (highlighted in blue above). The first listing is the number of timesteps in the tstab interval. The second listing is the number of timepoints in the shooting interval.

CPU time, elapsed (clock) time, and memory statistics are listed (highlighted in pink above). When APS is used with multi-threading, the CPU time is generally larger than the elapsed time. CPU time totals the usage of all CPUs.

Guidelines for Setting up Oscillators for Simulation

Beat frequency

Enter a frequency that is between 0.25 * actual frequency to 4 * actual frequency. As long as the specified frequency is within this range, the solution should be possible.

Tstab

Tstab is the time interval that is used to start the oscillator and allows the oscillations to build up to a near steady-state amplitude. *Decide automatically* is recommended for the *Run transient?* selection to automate the tstab setting.

Dynamic parameters are available during the tstab interval. Dynamic parameters allow changing option settings during tstab and are typically used to reduce the accuracy (which speeds up the simulation) at the beginning and then tighten the accuracy at the end to improve convergence.

Different ways to start the oscillator

The first thing to do when simulating oscillators is to reliably start the oscillator. Different methods are used for different types of oscillators. These are:

- **Moderate Q Feedback Oscillators**: If the oscillator is a feedback oscillator, then the linear oscillator initial condition (oscic) is highly recommended. When this is selected, the beat frequency needs to be within 0.5 * actual frequency to 1.5 * actual frequency. This performs a variation of the linear stability analysis after the DC analysis for the first timepoint in the tstab interval. It provides a very good estimate of the oscillating frequency and amplitude at every node and this estimate is used at the beginning of the tstab interval. This allows the duration of the tstab period to be about 5 to 10 periods of

© 1994-2020 All Rights Reserved.

the oscillator frequency for normal Q circuits. In many cases, the tstab time can be set to zero. Doing this, causes the pss analysis to start with the estimated frequency and amplitude from the linear oscic.

Another way to start the oscillator is to use a piecewise linear source as the power supply. At time zero, the voltage is about 80% to 90% of nominal. The second timepoint should be about one half the period of the oscillator and the full power supply voltage. The bump in voltage causes a voltage transient that starts things up. For differential oscillators, do this on one side of the differential system.

A third way is to add a current pulse to a point in the circuit that causes the oscillation to start using an ipulse component from *analogLib*. Specify a single pulse that is large enough to get things started. This is accomplished by adding an ipulse component from *analogLib* and specifying the zero value to zero, the one value to be about a milliamp, the rise and fall times to about one tenth of a period, and the pulse width to about three tenths of a period. Leave the period field blank to indicate that only a single pulse should be generated. The ipulse is an ideal open when the current is zero, so it does not load the circuit.

For all the different methods other than the linear oscic, tstab must be long enough for the oscillator to reach near steady-state operation. For most oscillators, this is about 100 periods of the oscillation frequency.

- **High Q Oscillators**: For high Q oscillators, harmonic balance is recommended. The reason for this is that in harmonic balance, the motional equivalent circuit (electrical representation of the crystal) goes directly into the frequency domain representation of the circuit. In shooting, there is always a small numerical integration error in the solution. For high Q circuits, this error causes uncertainty in the pss result that can cause the pss analysis to become inaccurate.
- **Ring Oscillators**: For ring oscillators, start the oscillation by specifying initial conditions at the output of one stage of the ring. Force the output of this stage to low or high with the initial conditions. This will force the oscillator into a known state at time zero, and then at the beginning of the transient, the node(s) will begin the switch to the other state, thereby starting the oscillator. Tstab should be set to about 1 or 2 periods.

For all the different methods other than the linear oscic, tstab must be long enough for the oscillator to reach near steady-state operation. For most oscillators, this is about 100 periods of the oscillation frequency. When using linear oscic, this can usually be shortened to about 10 periods or even zero.

As usual for any pss analysis, first the tstab interval is simulated. Then one period of the frequency supplied in the pss *Choosing Analyses* form is simulated. For oscillators, an additional four periods of the specified frequency in the pss setup is simulated. In the last interval, Spectre checks for possible divided-down frequencies. If these frequencies are

© 1994-2020 All Rights Reserved.

found, Spectre adjusts the frequency/period to correspond to the new lower frequency and tries to solve for that lower frequency.

Saveinit- A way to save the startup waveform

For many oscillators, it is desirable to see the actual startup waveform in order to verify that the startup is robust. To see the startup waveform, select *yes* for the *Save Initial Transient Results (saveinit)* property in the *Choosing Analyses* form. This will write the initial waveforms to the output data for viewing.

Oscillator Node- Used to estimate the frequency for the first pss iteration

When linear oscic is used, the oscillator node must be a node that is inside the feedback system for the oscillator. This is used to estimate the initial oscillating frequency. If the oscillator is differential, specify a pair of nodes in a symmetrical location in the oscillator.

Regardless of the setting for oscic, the oscillator node is used at the end of the tstab interval to determine an initial estimate of the actual oscillation frequency.

If the amplitude on this node drops to less than 1mV peak, an error is displayed indicating that the amplitude of the oscillator node is too small to properly estimate the frequency, and the simulation stops. If the frequency gets to 1000 * frequency specified, an error is displayed indicating that the frequency is too high to be reliable.

© 1994-2020 All Rights Reserved.

ADE Implementation

Below is an example ADE Explorer setup for an oscillator circuit. Let us set up a PSS analysis, run the simulation, and plot the output waveform.

Consider an oscillator shown below.

© 1994-2020
All Rights Reserved.

1. Select *Analyses - Choose* to open the *Choosing Analyses* form.

© 1994-2020
All Rights Reserved.

2. Specify an estimate of the oscillation frequency (highlighted in red above). If you use *linear* oscic, the estimate should be between 0.5 * actual oscillating frequency to 1.5 * actual oscillating frequency. Otherwise, it can be between 0.25 and 4 times the actual operating frequency.
3. Select *Conservative* (recommended) for oscillators (highlighted in green above).
4. Select the Calculate initial conditions automatically check box.
5. The oscillator node should be inside the feedback system if *linear* oscic is used (highlighted in blue above). If the *default* oscic is used, the node just needs a signal on it.
6. *Select linear* oscic (recommended) for feedback oscillators (highlighted in purple above).

   default oscic starts from the time zero timepoint in the tstab interval.

Note: For a driven circuit, the process is the same except that the oscillator node and the oscillator initial condition fields are not shown.

© 1994-2020 All Rights Reserved.

Spectre Output File (Oscillators)

```
/servers/scratch03/kamal/spectrerf131/testcase131/docs_datab
File  Edit  View  Help                                        cadence

Using linear IC
Linear IC: estimated frequency is 5.12282e+09 Hz

================================
`pss': time = (0 s -> 11.0508 ns)
================================

Important parameter values in tstab integration:
    start = 0 s
    outputstart = 0 s
    stop = 11.0508 ns
    period = 195.205 ps
    maxperiods = 50
    step = 10.27 ps
    maxstep = 8 ps
    ic = all
    useprevic = no
    skipdc = no
    reltol = 1e-03
    abstol(V) = 1 uV
    abstol(I) = 1 pA
    temp = 27 C
    tnom = 27 C
    tempeffects = all
    method = traponly
    lteratio = 3.5
    relref = sigglobal
    cmin = 0 F
    gmin = 1 pS

    pss: time = 282.2 ps    (2.55 %), step = 6.36 ps     (57.6 m%)
    pss: time = 836.6 ps    (7.57 %), step = 8 ps        (72.4 m%)
    pss: time = 1.384 ns    (12.5 %), step = 7.151 ps    (64.7 m%)
    pss: time = 1.941 ns    (17.6 %), step = 7.486 ps    (67.7 m%)
    pss: time = 2.489 ns    (22.5 %), step = 6.395 ps    (57.9 m%)
    pss: time = 3.045 ns    (27.6 %), step = 7.058 ps    (63.9 m%)
    pss: time = 3.593 ns    (32.5 %), step = 7.95 ps     (71.9 m%)
    pss: time = 4.147 ns    (37.5 %), step = 6.761 ps    (61.2 m%)
    pss: time = 4.702 ns    (42.6 %), step = 6.99 ps     (63.3 m%)
    pss: time = 5.251 ns    (47.5 %), step = 7.565 ps    (68.5 m%)
    pss: time = 5.806 ns    (52.5 %), step = 6.389 ps    (57.8 m%)
    pss: time = 6.355 ns    (57.5 %), step = 7.38 ps     (66.8 m%)
    pss: time = 6.91 ns     (62.5 %), step = 7.978 ps    (72.2 m%)
29  >                                                    L163    C29
```


© 1994-2020 All Rights Reserved.

The estimate of the oscillation frequency from the linear oscic (highlighted in red above) is shown first.

```
/servers/scratch03/kamal/spectrerf131/testcase131/docs_datab
File  Edit  View  Help                                          cadence
    pss: time = 9.67 ns      (87.5 %), step = 780.8 fs     (7.07 m%)
    pss: time = 10.22 ns     (92.5 %), step = 780.8 fs     (7.07 m%)
The Estimated oscillating frequency from Tstab Tran is = 5.15009 GHz .
Total time required for tstab analysis `pss': CPU = 1.48977 s, elapsed = 5.35928 s
Time accumulated: CPU = 1.49077 s, elapsed = 5.35937 s.
Peak resident memory used = 41.3 Mbytes.

Conv norm = 1e+03, max dV(I12.I20.a) = -3.53428 V, took 510 ms.

Important parameter values in pss iteration:
    start = 10.2278 ns
    outputstart = 0 s
    stop = 10.422 ns
    period = 194.171 ps
    maxperiods = 50
    steadyratio = 100e-03
    step = 10.27 ps
    maxstep = 485.429 fs
    ic = all
    useprevic = no
    skipdc = no
    reltol = 10e-06
    abstol(V) = 1 uV
    abstol(I) = 1 pA
    temp = 27 C
    tnom = 27 C
    tempeffects = all
    errpreset = conservative
    method = gear2only
    lteratio = 10
    relref = alllocal
    cmin = 0 F
    gmin = 1 pS


Notice from spectre during periodic steady state analysis `pss'.
    37 warnings suppressed.


=======================================
`pss': time = (10.2278 ns -> 10.422 ns)
=======================================
    pss: time = 10.23 ns     (2.73 %), step = 485.4 fs     (250 m%)
29  >                                                    L163    C29
```


© 1994-2020 All Rights Reserved.

The frequency from the end of the tstab interval (highlighted in red above) is displayed next.

```
/servers/scratch03/kamal/spectrerf131/testcase131/docs_datab
File  Edit  View  Help                                          cadence
    pss: time = 10.29 ns    (32.7 %), step = 485.4 fs     (250 m%)
    pss: time = 10.3 ns     (37.7 %), step = 485.4 fs     (250 m%)
    pss: time = 10.31 ns    (42.7 %), step = 485.4 fs     (250 m%)
    pss: time = 10.32 ns    (47.7 %), step = 485.4 fs     (250 m%)
    pss: time = 10.33 ns    (52.7 %), step = 485.4 fs     (250 m%)
    pss: time = 10.34 ns    (57.7 %), step = 485.4 fs     (250 m%)
    pss: time = 10.35 ns    (62.7 %), step = 485.4 fs     (250 m%)
    pss: time = 10.36 ns    (67.7 %), step = 485.4 fs     (250 m%)
    pss: time = 10.37 ns    (72.7 %), step = 485.4 fs     (250 m%)
    pss: time = 10.38 ns    (77.7 %), step = 485.4 fs     (250 m%)
    pss: time = 10.39 ns    (82.7 %), step = 485.4 fs     (250 m%)
    pss: time = 10.4 ns     (87.7 %), step = 485.4 fs     (250 m%)
    pss: time = 10.41 ns    (92.7 %), step = 485.4 fs     (250 m%)
    pss: time = 10.42 ns    (97.7 %), step = 485.4 fs     (250 m%)
Conv norm = 13.4e-03, max dI(V5:p) = -3.94583 nA, took 100 ms.

Fundamental frequency is 5.14985 GHz.

Opening the PSF file ../psf/pss.td.pss ...

Opening the PSF file ../psf/pss.fd.pss ...
Number of accepted pss steps =              3507
Initial condition solution time: CPU = 11.999 ms, elapsed = 11.9619 ms.
pss: The steady-state solution was achieved in 4 iterations.
Number of accepted pss steps =               403
Total time required for pss analysis `pss': CPU = 1.2838 s, elapsed = 1.35938 s.
Time accumulated: CPU = 1.87371 s, elapsed = 5.77917 s.
Peak resident memory used = 46.5 Mbytes.


Notice from spectre.
    27 notices suppressed.

modelParameter: writing model parameter values to rawfile.

Opening the PSF file ../psf/modelParameter.info ...
element: writing instance parameter values to rawfile.

Opening the PSF file ../psf/element.info ...
outputParameter: writing output parameter values to rawfile.

Opening the PSF file ../psf/outputParameter.info ...
designParamVals: writing netlist parameters to rawfile
29  >                                                   L163    C29
```

The solution frequency and the number of timesteps (highlighted in red above) are shown. The first listing of *accepted pss timesteps* is the number of timesteps in the tstab interval. The second number is the number in the shooting interval.

Dynamic Parameters in tstab

Simulator options can be changed at different times during the tstab interval using dynamic parameters. This is typically used to loosen convergence tolerances which makes the

© 1994-2020 All Rights Reserved.

simulation less accurate but faster at the beginning of tstab and then tightening the tolerances at the end to improve convergence. Single options are supported directly in the *Choosing Analyses* form.

Oscillator Tuning Mode

Oscillator tuning mode is provided to tune the oscillator to a desired frequency and then run the selected small-signal analyses. This is useful for sweeps and for Monte Carlo simulations. To run an oscillator tuning analysis, perform the following:

1. Open an oscillator circuit, and start ADE Explorer. Open the *Choosing Analyses* form, and select *pss* analysis.
2. Fill out the form as usual, except for the following:
   a. In the *Fundamental Frequency* field, type the desired frequency target.
   b. Click the check box next to *Enable tuning mode analysis*.

© 1994-2020 All Rights Reserved.

c. Select whether you want to tune a variable, temperature, or a component parameter. the example below shows a component parameter.

© 1994-2020
All Rights Reserved.

d. Enable the desired small-signal analyses.

e. Run the analyses. When the simulation completes, open the *Direct Plot Form* from ADE Explorer.

PSS Options

The options are divided into five tabs: *Convergence*, *Accuracy*, *Output*, *Reuse*, and *Misc* (miscellaneous).

The options are shown in the order that they occur in the options form in ADE Explorer.

© 1994-2020 All Rights Reserved.

Convergence tab

Ic (Shooting and Harmonic Balance)

Initial conditions can be specified graphically by selecting *Simulation* - *Convergence Aids* - *Initial Condition* in ADE Explorer. Initial conditions can also be specified from a file using the *readic* property. Capacitors and inductors have the initial condition properties in the property list for the component. For capacitors, this is an initial voltage across the capacitor, and for inductors, it is an initial current in the inductor. The default is to observe all the initial conditions in the DC analysis that is used as the time-zero timepoint. The initial conditions force a voltage or current to be present in the time-zero solution. The initial conditions are released for the rest of the transient simulation in the tstab interval. The *ic* option controls the initial conditions that should be observed in the time-zero timepoint. *all* is the default. *dev* means that only the initial conditions on capacitors and inductors are observed. *node* means

© 1994-2020 All Rights Reserved.

that only the initial conditions on a node are observed. *dc* means that no initial conditions are observed. The example below shows the ic option set to *node*.

Skipdc (Shooting and Harmonic Balance)

In some cases, the time-zero timepoint DC analysis does not converge. Instead of stopping the simulation, *skipdc* allows the tstab simulation to continue using an assumed solution for the time-zero timepoint. The default is *no* and a DC analysis is run to get the initial timepoint. *yes* means skip the DC solution, and proceed directly to the tstab simulation. All the nodes with initial conditions specified start at the initial condition value. Nodes with batteries start at the battery voltage. Nodes with no initial conditions start at zero volts. For `skipdc=yes`, the signal sources start as specified immediately in the tstab simulation. *sigrampup* uses the same assumptions for the starting voltages as *yes* does, but the start time is set to negative one tenth of the stop time for the tstab interval. At this time, the signal source time-varying part starts at zero and linearly ramps up to the full value at time = zero. After time = zero, the sources have the full amplitude time-varying part. The example below shows the *skipdc* option set to *sigrampup*.

Readic (Shooting and Harmonic Balance)

This specifies an ascii file that contains two columns to be read as initial conditions. The first column is the node name. The second column is the voltage value. If the entry does not start with / (slash), the entry is located in the netlist directory. To find the netlist directory, select *Setup- Simulator/Directory/Host* in ADE Explorer. Look in the *Project Directory* field for the location of the simulation directory. Navigate to that directory and then to the `<Circuit Name>/spectre/<schematic or config>/netlist` directory. You can also click (...) and browse to the directory. An example is shown below.

Readns (Shooting and Harmonic Balance)

This specifies an ascii file with the same format as an ic file that is used as nodesets for the time-zero DC solution. Nodesets do not force a voltage to be held for the time-zero solution. Instead, they are a way of speeding up the time-zero calculation. As a suggestion, set the

© 1994-2020 All Rights Reserved.

write option and the *readns* option to the same filename. The *write* option writes the time-zero solution to a file. When this is used as a starting point, many fewer iterations are needed for the time-zero point to converge. An example is shown below.

Cmin (Shooting and Harmonic Balance)

cmin can be used to improve convergence in the tstab and shooting intervals for shooting and for the tstab interval for harmonic balance. If a value is set for *cmin*, a capacitor with this value is added to every node with the other terminal of the capacitor connected to the global ground node. If a 10f to 50f capacitor is added, this prevents instantaneous changes from occurring from timepoint to timepoint, thus improving the convergence at the cost of adding non-physical capacitors to the circuit. An example with 10 femtoFarads is shown below. Note that if 10 is entered, a 10 Farad capacitor is added from every node to ground. Remember the multiplier in the entry. An example is shown below.

Maxiters (Shooting and Harmonic Balance)

The transient analysis for tstab (for both shooting and hb) and shooting interval iterate to a solution at each timepoint. *maxiters* specifies the maximum number of iterations before the timestep is cut for another try at convergence. In some cases, model parameters can cause discontinuities in the device current or capacitance. If this occurs, the change in the circuit condition can be large enough to require more than the five iterations that are the default. Specifying *maxiters* to the 40 to 100 range usually allows the simulator to converge in spite of the discontinuity. An example with *maxiters* set to 40 is shown below.

© 1994-2020 All Rights Reserved.

Accuracy Tab

The *Accuracy* tab is split into two categories: INTEGRATION METHOD PARAMETERS and ACCURACY PARAMETERS. For each parameter, the most commonly used settings are discussed.

© 1994-2020 All Rights Reserved.

© 1994-2020
All Rights Reserved.

Method (Shooting only)

This option controls the integration method for the shooting window. Note that harmonic balance does not have the *method* option because the solver works in the frequency domain. The default value for the shooting window is determined by the setting of *errpreset*. For *moderate* and *conservative*, the default is *gear2only*. For *liberal* it is *traponly*. Generally *gear2only* is preferred because of the absence of trapezoidal ringing inherent in the *trap* (trapezoidal) method. Note that *liberal* is not recommended for RF simulation.

In the case where you are simulating a crystal oscillator with shooting, the method option should be set to *traponly*. The ringing that is inherent in the trapezoidal method is present and may cause convergence failure because of the unpredictable amplitude of the ringing, which depends on the accumulated total numerical integration error. The benefit of having a method that neither damps nor emphasizes the oscillations is required when simulating high Q oscillators. Even though `trap` neither emphasizes nor damps the oscillations, the small numerical integration errors caused by the finite timestep can still cause some inaccuracy in the simulation result. Using *gear2only* is not recommended even though convergence is more likely because the numerical damping inherent in the *gear2only* method causes unacceptable damping in the resonator, which causes smaller than real amplitude inside the resonator. Because of this, the overall solution will result in an error and can cause inaccurate measurement of the phase noise.

Tstabmethod (Shooting and Harmonic Balance)

This option controls the integration method in the tstab interval. The default value is determined by the setting of *errpreset* and whether the circuit is driven or is an oscillator. For *moderate* and *conservative* with a driven circuit, the default is *gear2only*. For *liberal* it is *traponly*. Generally *gear2only* is preferred because of the absence of trapezoidal ringing inherent in the *trap* (trapezoidal) method. The default for *tstabmethod* is *traponly* for oscillators. *traponly* is used for oscillators because with the defaults, the tstab interval uses relatively loose convergence options in order to speed up the simulation. This causes relatively long timesteps which for the *gear2* method would cause noticeable numerical damping. *traponly* does not numerically damp, therefore, it provides a better estimate of the oscillating frequency and amplitude. An example with *tstabmethod* set to *gear2only* is shown below.

© 1994-2020 All Rights Reserved.

Relref (Shooting (tstab and shooting window) and Harmonic Balance (tstab only))

In the order of most to least accurate, the *relref* settings are *pointlocal*, *alllocal*, *sigglobal*, *allglobal*. The default for *moderate* and *conservative* is *alllocal* which is preferred. *relref* is used in the transient analysis in the tstab interval for both shooting and harmonic balance and in the shooting window for shooting. *relref* is used in hb only for the tstab interval because the solver works in the frequency domain. In some cases, the timestep can collapse to near zero. The symptom is that in the Spectre output window, the percent done number remains the same for many reporting intervals. The solution to this is to set *relref* to *sigglobal*, which is slightly less accurate. In the Fourier transform, the noise floor usually degrades by 3 to 6 dB. Another possibility is to use the *lteminstep* option as described in the *AdditionalParams* option of the *Misc* tab.

Refer to Relref on page 664 for a detailed discussion on *relref*.

An example with *relref* set to *sigglobal* is shown below.

Lteratio Shooting ((tstab and shooting window) and Harmonic Balance (tstab only))

lteratio is a multiplier for the allowable numerical integration error in the tstab interval and the shooting window. The default is 3.5 or 10 depending on the setting of *reltol* and *errpreset*. See Controlling Accuracy on page 672 for details. *lteratio* cannot be set smaller than 1.0 and is normally between 3.5 and 100. In some cases, the timestep collapses to near zero and setting *relref* to *sigglobal* does not fix the problem. In this case, there might be a model discontinuity. To test this, disable numerical integration timestep control by setting *lteratio* to 1e9. Note that this is an extreme measure not to be used under normal simulations. If the timestep still collapses, there is a discontinuity in one of the models in the circuit. This could be a device model or a Verilog-A model. If it is necessary to set *lteratio* very large to get the simulation to complete, you must also set *maxstep* small enough to preserve the accuracy of the simulation. An example with *lteratio* set to the smallest value of *1.0* is shown below. Note that *lteminstep* might also be used to treat this condition. See the AdditionalParams (Shooting and Harmonic Balance) on page 748 section in the *Misc* tab on how to use this option.

© 1994-2020 All Rights Reserved.

Steadyratio (Shooting Only)

The maximum mismatch between the beginning and ending values for the voltages and currents in the shooting window is determined by the *convergence criteria * Iteratio * steadyratio*. In some cases, with oscillators, *conv_norm* drops to a small value and then bumps up and down. However, it never gets less than one, therefore, the convergence fails. In this case, setting *steadyratio* to 1.0 will cause convergence if the *conv_norm* number gets to 10 or less. While larger settings of *steadyratio* may cause the oscillator to converge, more mismatch from beginning to end of the shooting window is allowed, which allows much more startup behavior to be in the solution. Setting *steadyratio* larger than one is not recommended. An example with *steadyratio* set to 1.0 is shown below.

Steadyratio (Harmonic Balance Only)

steadyratio in harmonic balance is used as a multiplier for the maximum delta that is allows from iteration to iteration. The default is 1.0. In some cases, *resd_norm* falls below one which indicates that the KCL check at each node is passing, but the Delta Norm is greater than one, which indicates that the motion from iteration-to-iteration is larger than the maximum allowed. Only in this case should *steadyratio* be set larger than one. Set it just larger than the value of the *resd_norm*. In this case, the smaller harmonics are changing relatively a lot, but because they are small, they do not affect the accuracy of the solution.

Itres (Shooting Only)

Use the default for this option. *itres* controls the precision (number of digits solved for) in the first iteration. The default is 1e-4 which causes 4 digits of resolution in the first iteration. Subsequent iterations are solved with more precision.

Itres (Harmonic Balance Only)

The default for *itres* for harmonic balance is 0.9, which allows up to 90% difference from the exact solution on the first iteration. Although this seems large, solving exactly for the first iteration can frequently cause non-convergence. As the iterations progress, the difference is reduced so that at the solution, full accuracy is achieved. In the case of very linear systems like an amplifier, tightening *itres* to the 1e-2 to 1e-4 range can reduce the number of frequency domain iterations and runtime. Also, when transient assist is used, the fft provides an excellent starting point for the first iteration. Therefore, setting *itres* to the 1e-2 to 1e-4 range can also reduce the number of frequency domain iterations and the runtime.

© 1994-2020 All Rights Reserved.

Resgmrescycle (Shooting and Harmonic Balance)

Leave this option at the default.

Inexact Newton (Shooting Only)

In the simulation time between iterations, an estimate is made for the starting point for the next iteration (shooting) or the harmonic content for the next iteration (hb). *inexactNewton* causes an inexact solution for the matrix that calculates the next starting point to be calculated. In some cases, solving the matrix exactly on the first iteration can cause the iterative solver to need many iterations to achieve convergence. Setting i*nexactNewton* causes a different series of iterations, which may speed up the simulation. The inexactness is reduced as the iterations progress. Try this option when the time spent between iterations is excessive.

Maxacfreq (Shooting Only)

If maximum sideband in *pac*, *pnoise*, *pxf*, or *psp* is set larger than 40, then the *pss* option *maxacfreq* needs to be set to the pss beat frequency times the largest maximum sideband number in the small-signal analyses. For example, if *pac* has 20 for maximum sidebands, *pnoise* has 70, *pxf* has 10, and *psp* has 50, then *maxacfreq* needs to be set to the pss frequency times 70.

Spectre may issue a warning at the beginning of the small-signal analysis like the following one from pnoise if *maxacfreq* is not set and more than 40 sidebands are specified. If you see this warning in a small-signal analysis, set the pss option *maxacfreq* to the pss beat

© 1994-2020 All Rights Reserved.

frequency times the number of sidebands used in the small-signal analysis. In the example below, *maxacfreq* should be set to 150 * pss frequency.

```
/net/lnx-rdavis3/usr2/rdavis/demos/ic61/docs_database/simulation/DVCO_RFX_tb1/spectre/schematic/psf/spectre.out
File  Help                                                                                                  cadence

*****************************************************************************
Periodic Noise Analysis `pnoise': freq = 5.14989 GHz + (1 kHz -> 50 MHz)
*****************************************************************************
Using the operating-point information generated by PSS analysis `pss'.

Warning from spectre during PNoise analysis `pnoise'.
    WARNING (SPCRTRF-15029): The following results might not be accurate due to undersampling of the actual frequencies above
        Lower side band of 150th harmonic for all stimulus frequencies.
        Lower side band of 149th harmonic for all stimulus frequencies.
        Lower side band of 148th harmonic for all stimulus frequencies.
        Lower side band of 147th harmonic for all stimulus frequencies.
        Lower side band of 146th harmonic for all stimulus frequencies.
        Further occurrences of this warning will be suppressed.

The estimated line width of the oscillator is 1.23919 kHz.
    pnoise: freq = 1.778 kHz   (5.26 %), step = 778.3 Hz     (5.26 %)
    pnoise: freq = 3.162 kHz   (10.5 %), step = 1.384 kHz    (5.26 %)
    pnoise: freq = 5.623 kHz   (15.8 %), step = 2.461 kHz    (5.26 %)
    pnoise: freq = 10 kHz      (21.1 %), step = 4.377 kHz    (5.26 %)
    pnoise: freq = 17.78 kHz   (26.3 %), step = 7.783 kHz    (5.26 %)
    pnoise: freq = 31.62 kHz   (31.6 %), step = 13.84 kHz    (5.26 %)
    pnoise: freq = 56.23 kHz   (36.8 %), step = 24.61 kHz    (5.26 %)
    pnoise: freq = 100 kHz     (42.1 %), step = 43.77 kHz    (5.26 %)
    pnoise: freq = 177.8 kHz   (47.4 %), step = 77.83 kHz    (5.26 %)
    pnoise: freq = 316.2 kHz   (52.6 %), step = 138.4 kHz    (5.26 %)
    pnoise: freq = 562.3 kHz   (57.9 %), step = 246.1 kHz    (5.26 %)
    pnoise: freq = 1 MHz       (63.2 %), step = 437.7 kHz    (5.26 %)
    pnoise: freq = 1.778 MHz   (68.4 %), step = 778.3 kHz    (5.26 %)
    pnoise: freq = 3.162 MHz   (73.7 %), step = 1.384 MHz    (5.26 %)
    pnoise: freq = 5.623 MHz   (78.9 %), step = 2.461 MHz    (5.26 %)
    pnoise: freq = 10 MHz      (84.2 %), step = 4.377 MHz    (5.26 %)
    pnoise: freq = 17.78 MHz   (89.5 %), step = 7.783 MHz    (5.26 %)
    pnoise: freq = 31.62 MHz   (94.7 %), step = 13.84 MHz    (5.26 %)
31  HelpAction
```

An example of *maxacfreq* set to 120 GHz is shown below.

```
maxacfreq    120G
```

Maxperiods (Shooting and Harmonic Balance)

Any iterative method may fail to converge, and therefore needs an iteration limit. Maxperiods sets the maximum number of times the waveform in the shooting window can be simulated or the number of hb iterations. The default is 20 times for a driven circuit, and 50 times for an oscillator in shooting, and 100 for hb. This is a minimum allowed value for this option. Setting a number less than this default will be ignored by Spectre. This should be set higher only after watching the conv_norm during the pss iterations. If the conv_norm is trending down with no sudden large increase, setting more iterations may allow convergence. If the conv_norm

© 1994-2020 All Rights Reserved.

trends down, and then gets large quickly, setting more iterations is very unlikely to help. An example with 100 is shown below.

Finitediff (Shooting Only)

Leave this option at the default. This is set automatically when *readpss* is set. It allows small changes in circuit values when the *readpss* option is used.

Highorder (Shooting Only)

The normal integration method in the shooting window is *gear2only* which is a second order integration method. *highorder* allows the integration method to have up to 16th order numerical integration, thus making the numerical noise floor of the simulation much smaller. When *highorder* is set to *yes*, a normal pss is run until convergence is achieved, and then that solution is refined using a higher order integration method. In practice this has value for very linear systems like LNAs, and has smaller value for more complicated blocks like mixers. When *highorder* is set to *yes*, there can be considerable time when the highorder algorithm is running, but there is no output to the spectre output window. This is normal for highorder. An example of `highorder=yes` is shown below.

© 1994-2020 All Rights Reserved.

Below is the runtime for `highorder = no`.

```
/servers/scratch03/kamal/spectrerf131/testcase131/docs_databas
File  Edit  View  Help                                          cadence
    pss: time = 14.99 ns     (57.5 %), step = 40 fs          (20 m%)
    pss: time = 15 ns        (62.5 %), step = 40 fs          (20 m%)
    pss: time = 15.01 ns     (67.5 %), step = 40 fs          (20 m%)
    pss: time = 15.02 ns     (72.5 %), step = 40 fs          (20 m%)
    pss: time = 15.03 ns     (77.5 %), step = 40 fs          (20 m%)
    pss: time = 15.04 ns     (82.5 %), step = 40 fs          (20 m%)
    pss: time = 15.05 ns     (87.5 %), step = 40 fs          (20 m%)
    pss: time = 15.06 ns     (92.5 %), step = 40 fs          (20 m%)
    pss: time = 15.07 ns     (97.5 %), step = 40 fs          (20 m%)
Conv norm = 346e-03, max dI(I23:bal_p_flow) = -77.2919 nA, took 1.72 s.

Fundamental frequency is 5 GHz.
Parameter V3:dc is 9.7501e-01.

Opening the PSF file ../psf/pss.td.pss ...

Opening the PSF file ../psf/pss.fd.pss ...

Opening the PSF file ../psf/pss.osctp.pss ...
Number of accepted pss steps =              43201
Initial condition solution time: CPU = 7.999 ms, elapsed = 8.60095 ms.
pss: The steady-state solution was achieved in 5 iterations.
Number of accepted pss steps =              5004
Total time required for pss analysis `pss': CPU = 8.78266 s, elapsed = 8.97707 s.
Time accumulated: CPU = 8.97863 s, elapsed = 9.40523 s.
Peak resident memory used = 75.8 Mbytes.


Notice from spectre.
    347 notices suppressed.

modelParameter: writing model parameter values to rawfile.

Opening the PSF file ../psf/modelParameter.info ...
element: writing instance parameter values to rawfile.

Opening the PSF file ../psf/element.info ...
outputParameter: writing output parameter values to rawfile.

Opening the PSF file ../psf/outputParameter.info ...
designParamVals: writing netlist parameters to rawfile.

Opening the PSFASCII file ../psf/designParamVals.info
21  >                                                   L577    C62
```


© 1994-2020 All Rights Reserved.

Below is the runtime and the report of the order that was used in the simulation when *highorder* was set to *yes*. The runtime is much longer.

```
/servers/scratch03/kamal/spectrerf131/testcase131/docs_database/si
File  Edit  View  Help                                                cādence
    pss: time = 15.02 ns    (72.5 %), step = 40 fs         (20 m%)
    pss: time = 15.03 ns    (77.5 %), step = 40 fs         (20 m%)
    pss: time = 15.04 ns    (82.5 %), step = 40 fs         (20 m%)
    pss: time = 15.05 ns    (87.5 %), step = 40 fs         (20 m%)
    pss: time = 15.06 ns    (92.5 %), step = 40 fs         (20 m%)
    pss: time = 15.07 ns    (97.5 %), step = 40 fs         (20 m%)
Conv norm = 729e-06, max dI(V3:p) = -171.73 pA, took 1.72 s.

Order 12 used in     126 subintervals.

Conv residual norm = 20.6e-03.
Conv solution-change norm in fdtd = 335e-03.
Conv Norm for Pss Period = 5.34955182556439e-07

Conv residual norm = 11.1e-06.
Conv solution-change norm in fdtd = 116e-06.
Conv Norm for Pss Period = 2.93883211678939e-11
Number of refinements using multi-interval Chebyshev polynomial spectral algorithm = 1. T

Fundamental frequency is 5 GHz.


MIC-PSA finite-difference refinement finished, took 1.15 s.


Opening the PSF file ../psf/pss.td.pss ...

Opening the PSF file ../psf/pss.fd.pss ...

Opening the PSF file ../psf/pss.osctp.pss ...
Number of accepted pss steps =             1512
Initial condition solution time: CPU = 7.998 ms, elapsed = 9.05681 ms.
pss: The steady-state solution was achieved in 6 iterations.
Number of accepted pss steps =             1512
Total time required for pss analysis `pss': CPU = 15.3807 s, elapsed = 15.4743 s.
Time accumulated: CPU = 15.5686 s, elapsed = 15.9363 s.
22  >                                                          L637    C62
```


© 1994-2020 All Rights Reserved.

Below is a spectral comparison between standard pss and *highorder* set to *yes*. The convergence options are at the default and *conservative* is selected. The results are for an LNA. The numerical noise floor at 1GHz dropped from -148dBm to -292 dBm.

© 1994-2020
All Rights Reserved.

Below is a comparison of runtimes for an LNA-Mixer circuit. The first is with *highorder* set to *no*, and the second is with *highorder* set to *yes*.

```
/servers/scratch03/kamal/spectrerf131/testcase131/docs_datab:
File  Edit  View  Help                                          cādence

    pss: time = 49.25 ns     (92.5 %), step = 2.862 ps     (28.6 m%)
    pss: time = 49.76 ns     (97.5 %), step = 3.333 ps     (33.3 m%)
Conv norm = 5.84e-03, max dI(I0.I28.I6.ls1:1) = 9.81515 pA, took 1.46 s.


Opening the PSF file ../psf/pss.td.pss ...

Opening the PSF file ../psf/pss.fd.pss ...
Number of accepted pss steps =              16443
Initial condition solution time: CPU = 82.987 ms, elapsed = 83.729 ms.
pss: The steady-state solution was achieved in 3 iterations.
Number of accepted pss steps =              3125
Total time required for pss analysis `pss': CPU = 7.72183 s, elapsed = 7.96265 s.
Time accumulated: CPU = 8.25274 s, elapsed = 9.00113 s.
Peak resident memory used = 106 Mbytes.

modelParameter: writing model parameter values to rawfile.

Opening the PSF file ../psf/modelParameter.info ...
element: writing instance parameter values to rawfile.

Opening the PSF file ../psf/element.info ...
outputParameter: writing output parameter values to rawfile.

Opening the PSF file ../psf/outputParameter.info ...
designParamVals: writing netlist parameters to rawfile.

Opening the PSFASCII file ../psf/designParamVals.info ...
primitives: writing primitives to rawfile.

Opening the PSFASCII file ../psf/primitives.info.primitives ...
subckts: writing subcircuits to rawfile.

Opening the PSFASCII file ../psf/subckts.info.subckts ...

Aggregate audit (2:04:20 PM, Thur Jul 14, 2016):
Time used: CPU = 8.31 s, elapsed = 9.13 s, util. = 91.1%.
Time spent in licensing: elapsed = 624 ms, percentage of total = 6.84%.
Peak memory used = 106 Mbytes.
Simulation started at: 2:04:11 PM, Thur Jul 14, 2016, ended at: 2:04:20 PM, Thur Ju
spectre completes with 0 errors, 3 warnings, and 7 notices.

37  >                                                    L322    C60
```


© 1994-2020 All Rights Reserved.

/servers/scratch03/kamal/spectrerf131/testcase131/docs_databas

File Edit View Help cadence

```
    pss: time = 46.25 ns    (62.5 %), step = 3.333 ps    (33.3 m%)
    pss: time = 46.75 ns    (67.5 %), step = 3.333 ps    (33.3 m%)
    pss: time = 47.25 ns    (72.5 %), step = 3.333 ps    (33.3 m%)
    pss: time = 47.75 ns    (77.5 %), step = 3.333 ps    (33.3 m%)
    pss: time = 48.25 ns    (82.5 %), step = 3.219 ps    (32.2 m%)
    pss: time = 48.75 ns    (87.5 %), step = 3.333 ps    (33.3 m%)
    pss: time = 49.25 ns    (92.5 %), step = 2.862 ps    (28.6 m%)
    pss: time = 49.76 ns    (97.5 %), step = 3.333 ps    (33.3 m%)
Conv norm = 5.84e-03, max dI(I0.I28.I6.1s1:1) = 9.81515 pA, took 1.55 s.

Order 14 used in     169 subintervals.
Order 16 used in     418 subintervals.

Conv residual norm = 177.
Conv solution-change norm in fdtd = 491.

Conv residual norm = 5.76e-03.
Conv solution-change norm in fdtd = 25.6e-03.
Number of refinements using multi-interval Chebyshev polynomial spectral algorithm =

MIC-PSA finite-difference refinement finished, took 15.29 s.


Opening the PSF file ../psf/pss.td.pss ...

Opening the PSF file ../psf/pss.fd.pss ...
Number of accepted pss steps =             9054
Initial condition solution time: CPU = 31.995 ms, elapsed = 33.2322 ms.
pss: The steady-state solution was achieved in 3 iterations.
Number of accepted pss steps =             9054
Total time required for pss analysis `pss': CPU = 35.8156 s, elapsed = 36.3207 s.
Time accumulated: CPU = 36.0165 s, elapsed = 36.7894 s.
Peak resident memory used = 339 Mbytes.

modelParameter: writing model parameter values to rawfile.

Opening the PSF file ../psf/modelParameter.info ...
element: writing instance parameter values to rawfile.

Opening the PSF file ../psf/element.info ...
outputParameter: writing output parameter values to rawfile.
```

36 > L1 C1

© 1994-2020 All Rights Reserved.

The runtime is much longer for `highorder=yes`, however, it allows the intermodulation products to be resolved. *highorder* set to *yes* is shown in the bottom subwindow below.

Psaratio (Shooting Only)

This is active when *highorder* is set to *yes*. The default is *one*. Setting smaller values produce more accurate solutions at the expense of runtime. Normally, the default should be used. In the example below, *psaratio* is set to 0.2 and a comparison of the numerical noise floor at 1GHz is made for the LNA used above. The noise floor dropped about 8 dBm when *psaratio* is lowered.

© 1994-2020
All Rights Reserved.

Maxorder (Shooting Only)

Leave this option at the default of 16. The order is set automatically in the highorder algorithm.

This is active when *highorder* is set to *yes*. This controls the maximum order used for the highorder method.

Fullpssvec (Shooting Only)

Leave this option at the default. At the end of each pss iteration, the settled steady-state point needs to be estimated. Because of the inherently variable timestep, not all the timepoints in the pss solution need to be used to estimate the next starting point. Setting *fullpssvec* to *yes* causes all the timepoints to be read in to calculate the starting voltages for the next iteration.

© 1994-2020
All Rights Reserved.

Fdharms (Shooting Only)

fdharms controls the number of harmonics of the pss beat frequency that are exposed to the `nport`, `mtline`, and `tline` components. This should be set to the number of harmonics that are set for maximum harmonics in the pss *Choosing Analyses* form.

Output Tab

© 1994-2020
All Rights Reserved.

Annotate (Shooting and Harmonic Balance)

This option controls the level of detail in the output log. The default is *status*. *no* produces no output. *title* produces only the conv_norm number at each iteration. *sweep* counts down from 10 to zero at each iteration and displays conv_norm. *status* provides output at 2.5% and then at every 5% of the way to the end at each iteration and the conv_norm number. *steps* provides output at each timestep in the simulation and the conv_norm number at each iteration. An example is shown below with *title* selected.

© 1994-2020
All Rights Reserved.

Below is the Spectre output with *title* selected.

```
/servers/scratch03/kamal/spectrerf131/testcase131/dc
File  Edit  View  Help                                   cadence

    rabsshort = 1 mOhm

=========================================
`pss': time = (40.0019 ns -> 50.0019 ns)
=========================================
Conv norm = 2.91e+03, max dI(I0.I28.I6.ls1:1) = 4.89167 uA, took 1.58 s.


=========================================
`pss': time = (40.0019 ns -> 50.0019 ns)
=========================================
Conv norm = 5.84e-03, max dI(I0.I28.I6.ls1:1) = 9.81515 pA, took 1.53 s.

Opening the PSF file ../psf/pss.td.pss ...

Opening the PSF file ../psf/pss.fd.pss ...
Number of accepted pss steps =              16443
Initial condition solution time: CPU = 30.995 ms, elapsed = 31.6379 ms.
pss: The steady-state solution was achieved in 3 iterations.
Number of accepted pss steps =              3125
Total time required for pss analysis `pss': CPU = 7.09792 s, elapsed =
Time accumulated: CPU = 7.30589 s, elapsed = 7.6525 s.
Peak resident memory used = 106 Mbytes.

modelParameter: writing model parameter values to rawfile.

Opening the PSF file ../psf/modelParameter.info ...
element: writing instance parameter values to rawfile.

Opening the PSF file ../psf/element.info ...
outputParameter: writing output parameter values to rawfile.

Opening the PSF file ../psf/outputParameter.info ...
designParamVals: writing netlist parameters to rawfile.

Opening the PSFASCII file ../psf/designParamVals.info ...
primitives: writing primitives to rawfile.

48  >                                                    L280    C60
```


© 1994-2020 All Rights Reserved.

Below is the output with *sweep* selected.

```
/servers/scratch03/kamal/spectrerf131/testcase131/docs
File  Edit  View  Help                                   cādence
    tnom = 27 C
    tempeffects = all
    errpreset = conservative
    method = gear2only
    lteratio = 3.5
    relref = alllocal
    cmin = 0 F
    gmin = 1 pS
    rabsshort = 1 mOhm

========================================
`pss': time = (40.0019 ns -> 50.0019 ns)
========================================
......9......8......7......6......5......4......3......2......1......0
Conv norm = 2.91e+03, max dI(I0.I28.I6.ls1:1) = 4.89167 uA, took 1.67 s.


========================================
`pss': time = (40.0019 ns -> 50.0019 ns)
========================================
......9......8......7......6......5......4......3......2......1......0
Conv norm = 5.84e-03, max dI(I0.I28.I6.ls1:1) = 9.81515 pA, took 1.63 s.

Opening the PSF file ../psf/pss.td.pss ...

Opening the PSF file ../psf/pss.fd.pss ...
Number of accepted pss steps =              16443
Initial condition solution time: CPU = 33.995 ms, elapsed = 35.0838 ms.
pss: The steady-state solution was achieved in 3 iterations.
Number of accepted pss steps =              3125
Total time required for pss analysis `pss': CPU = 7.52486 s, elapsed = 7.
Time accumulated: CPU = 7.73182 s, elapsed = 8.08866 s.
Peak resident memory used = 106 Mbytes.

modelParameter: writing model parameter values to rawfile.

Opening the PSF file ../psf/modelParameter.info ...
element: writing instance parameter values to rawfile.

Opening the PSF file ../psf/element.info
49  >                                                    L283    C60
```


© 1994-2020
All Rights Reserved.

Below is the output with status *selected*.

```
/servers/scratch03/kamal/spectrerf131/testcase131/d
File  Edit  View  Help                                                cadence
    pss: time = 44.25 ns    (42.5 %), step = 3.333 ps    (33.3 m%)
    pss: time = 44.75 ns    (47.5 %), step = 3.333 ps    (33.3 m%)
    pss: time = 45.25 ns    (52.5 %), step = 3.333 ps    (33.3 m%)
    pss: time = 45.75 ns    (57.5 %), step = 3.333 ps    (33.3 m%)
    pss: time = 46.25 ns    (62.5 %), step = 3.333 ps    (33.3 m%)
    pss: time = 46.75 ns    (67.5 %), step = 3.333 ps    (33.3 m%)
    pss: time = 47.25 ns    (72.5 %), step = 3.333 ps    (33.3 m%)
    pss: time = 47.75 ns    (77.5 %), step = 3.333 ps    (33.3 m%)
    pss: time = 48.25 ns    (82.5 %), step = 3.219 ps    (32.2 m%)
    pss: time = 48.75 ns    (87.5 %), step = 3.333 ps    (33.3 m%)
    pss: time = 49.25 ns    (92.5 %), step = 2.862 ps    (28.6 m%)
    pss: time = 49.76 ns    (97.5 %), step = 3.333 ps    (33.3 m%)
Conv norm = 5.84e-03, max dI(I0.I28.I6.ls1:1) = 9.81515 pA, took 1.42 s

Opening the PSF file ../psf/pss.td.pss ...

Opening the PSF file ../psf/pss.fd.pss ...
Number of accepted pss steps =              16443
Initial condition solution time: CPU = 29.995 ms, elapsed = 30.5572 ms.
pss: The steady-state solution was achieved in 3 iterations.
Number of accepted pss steps =              3125
Total time required for pss analysis `pss': CPU = 6.68398 s, elapsed =
Time accumulated: CPU = 6.88695 s, elapsed = 7.22685 s.
Peak resident memory used = 106 Mbytes.

modelParameter: writing model parameter values to rawfile.

Opening the PSF file ../psf/modelParameter.info ...
element: writing instance parameter values to rawfile.

Opening the PSF file ../psf/element.info ...
outputParameter: writing output parameter values to rawfile.

Opening the PSF file ../psf/outputParameter.info ...
designParamVals: writing netlist parameters to rawfile.

Opening the PSFASCII file ../psf/designParamVals.info ...
primitives: writing primitives to rawfile.

Opening the PSFASCII file ../psf/primitives.info.primitives ...
subckts: writing subcircuits to rawfile
50  >                                                        L322   C60
```


© 1994-2020 All Rights Reserved.

Below is the output with *steps* selected.

```
/servers/scratch03/kamal/spectrerf131/testcase131/d
File  Edit  View  Help                                                cadence
    pss: time = 49.98 ns    (99.8 %), step = 3.333 ps    (33.3 m%)
    pss: time = 49.99 ns    (99.9 %), step = 3.333 ps    (33.3 m%)
    pss: time = 49.99 ns    (99.9 %), step = 3.333 ps    (33.3 m%)
    pss: time = 49.99 ns    (99.9 %), step = 3.333 ps    (33.3 m%)
    pss: time = 50 ns        (100 %), step = 3.333 ps    (33.3 m%)
    pss: time = 50 ns        (100 %), step = 2.164 ps    (21.6 m%)
    pss: time = 50 ns        (100 %), step = 2.164 ps    (21.6 m%)
Conv norm = 5.84e-03, max dI(I0.I28.I6.ls1:1) = 9.81515 pA, took 1.56 s.

Opening the PSF file ../psf/pss.td.pss ...

Opening the PSF file ../psf/pss.fd.pss ...
Number of accepted pss steps =              16443
Initial condition solution time: CPU = 29.995 ms, elapsed = 31.0371 ms.
pss: The steady-state solution was achieved in 3 iterations.
Number of accepted pss steps =              3125
Total time required for pss analysis `pss': CPU = 7.33788 s, elapsed =
Time accumulated: CPU = 7.54485 s, elapsed = 7.89519 s.
Peak resident memory used = 106 Mbytes.

modelParameter: writing model parameter values to rawfile.

Opening the PSF file ../psf/modelParameter.info ...
element: writing instance parameter values to rawfile.

Opening the PSF file ../psf/element.info ...
outputParameter: writing output parameter values to rawfile.

Opening the PSF file ../psf/outputParameter.info ...
designParamVals: writing netlist parameters to rawfile.

Opening the PSFASCII file ../psf/designParamVals.info ...
primitives: writing primitives to rawfile.

Opening the PSFASCII file ../psf/primitives.info.primitives ...
subckts: writing subcircuits to rawfile.

Opening the PSFASCII file ../psf/subckts.info.subckts ...

Aggregate audit (9:04:40 AM, Fri Jul 15, 2016):
Time used: CPU = 7.6 s, elapsed = 8.01 s, util. = 95%
51  >                                                    L16,710   C60
```


© 1994-2020 All Rights Reserved.

Save (Shooting and Harmonic Balance)

The default is *allpub*. This saves all the public voltages at all levels of the hierarchy in the schematic. It excludes the internal nodes of the device models. *all* adds the internal nodes of all the devices. *lvl* saves all the nodes including the internal nodes of the devices through the level of hierarchy set in the *nestlvl* option. *lvlpub* is like *lvl*, but it does not save the internal nodes. *selected* saves only the nodes that are specifically saved. In ADE Explorer, this is accomplished using the *Outputs - To Be Saved - Select On Schematic* and then selecting the nodes and terminals specifically in the schematic. At the netlist level, this is accomplished by using a save statement with a list of names to be saved.

Nestlvl (Shooting and Harmonic Balance)

If *save* is set to *lvl* or *lvlpub*, this controls the maximum level of hierarchy to be saved. If *nestlvl* is *1*, only the top level is saved. If *nestlvl* is 2, the top level and the next level down are saved. The value for *nestlvl* can be any integer. The example below shows *lvlpub* selected that saves two levels of hierarchy.

Compression (Shooting and Harmonic Balance)

Normally, this should not be used for RF simulation. It is not digital compression. For RF simulation, where the input is sinusoidal, the size of the results file will normally double. It is useful only for circuits that are predominantly square wave.

Oppoint (Shooting and Harmonic Balance)

Normally, this should be left at the default of *rawfile*. It controls where the time-zero operating point solution should be saved to.

Skipstart (Shooting and Harmonic Balance)

Normally, this is not used for RF simulation unless there is a very long tstab. It is a way of reducing the amount of data in the output file from the tstab interval by not writing many of the timepoints to the output file. Skipstart controls the simulation time where the skipping is to start. The default is 0 (zero).

© 1994-2020 All Rights Reserved.

Skipstop (Shooting and Harmonic Balance)

Normally, this is not used for RF simulation unless there is a very long tstab. This specifies the simulation time in the tstab interval where the skipping is to stop. The default is the stop time in the tstab interval.

Skipcount (Shooting and Harmonic Balance)

If *skipstart* and *skipstop* are set either *skipcount* or *strobeperiod* should be set, but not both. *skipcount* saves one out of every skipcount points to the output data file. A *skipcount* of 3 saves one, then skips two points. Skipcount of 10 saves one in 10 timepoints to the output file. The default is 1, which saves every point. An example is shown below. If *tstab* is 1u, the first 5% and the last 5% has all the datapoints saved, and in between, one out of ten points is saved.

Strobeperiod (Shooting and Harmonic Balance)

If *skipstart* and *skipstop* are set either *skipcount* or *strobeperiod* should be set, but not both. *strobeperiod* forces simulation datapoints between *skipstart* and *skipstop* at the interval of *strobeperiod*. Below is an example where the first 5% and the last 5% have all the datapoints, and in between there are timepoints at 0.01usecond intervals. Note that the simulator may actually calculate many points between the *skipstart* and *skipstop* times, but only the data at the strobe interval is saved to the output file.

Strobedelay (Shooting and Harmonic Balance)

strobedelay specifies a time after the skipstart point for the beginning of the strobing. *strobedelay* must be between 0 (zero) and the time set in *strobeperiod*.

© 1994-2020 All Rights Reserved.

Outputtype (Shooting Only)

This controls what type of information is to be saved in the output file. The default of *all* saves both the time and frequency data, but the data can be restricted to either time or frequency domain data.

Reuse Tab

© 1994-2020
All Rights Reserved.

Write (Shooting Only)

write specifies a filename to which the DC solution used as the first timepoint be written to. If the name does not start with slash (/), the file will be written in the *netlist* directory. To determine the netlist directory location, select *Setup - Simulator/Directory/Host* in ADE Explorer. The *Project Directory* field lists the location of the simulation directory. From this directory, the netlist directory is in `<Circuit Name>`/`<simulator name>`/*<schematic or config>/netlist*.

An example is shown below.

Writefinal (Shooting Only)

writefinal writes the final pss timepoint from the shooting window (not the final timepoint of the tstab interval) to the filename specified. See the description for *write* for the location. An example is shown below.

Swapfile (Shooting Only)

During the shooting interval, the solution matrices need to be saved in order to calculate the settled voltages of every node for the next iteration. The solution matrices are the internal solution for all the simultaneous equations for the circuit. The solution vector is also saved, and this is the solution voltages and currents. Normally, this data is kept in memory which makes the Spectre process get larger, starting with the second iteration of pss. On 32-bit systems, a maximum process-size of slightly less than 4GB limits the size of the circuit that can be simulated. Even on 64-bit systems, there is a limit set by the amount of memory that is installed on the system that is running Spectre.

Swapping that is done by the operating system is inefficient because there is no logical relationship for the swapped memory pages. Setting swapfile to a name causes Spectre to write all the solution matrices to the disk in sequential cylinders so that the read time from the disk is faster than using swapping by the operating system. Disk accesses are inherently much slower than memory accesses, however, using swapfile is considerably faster than swapping in the OS. Using swapfile reduces the size of the Spectre process so that it fits in the memory installed in the machine, or allows larger circuits to be simulated. There is a large amount of data that typically is written to the disk. Because of this, the best choice is to write the data in a disk that is local to the machine. If the data is written over a high-latency network,

© 1994-2020 All Rights Reserved.

it might actually be better to use the local swap space. Only a trial can determine which is better. Below is an example of setting *swapfile*.

swapfile writes to the netlist directory by default. To locate the simulation directory, select *Setup - Save Options* and look in the *Simulator Results Directory Location* field. Navigate to that directory, and then navigate to the *<`Circuit_Name`>/spectre/ <schematic_or_config>/netlist* directory. If the size of the swapfile gets larger than 1GB, it splits into multiple files, as shown below.

Writepss (Shooting and Harmonic Balance)

This option specifies that the full internal state of the pss analysis be written out to the file specified in the option. An example is shown below.

For large circuits, the file can be quite large.

© 1994-2020 All Rights Reserved.

Readpss (Shooting and Harmonic Balance)

This option specifies the file to be read in as a starting point for the next pss analysis. If nothing is changed in the circuit, then only one iteration is performed and small-signal analyses can be run with a much faster time for the pss analysis. Changes can be made to the circuit or analysis options as long as the topology of the circuit stays the same. Changes introduce a discontinuity at the beginning of the pss analysis, which might be large enough to cause the pss to not converge.

An example of *readpss* is shown below.

© 1994-2020 All Rights Reserved.

When a circuit value is changed, a refinement is done automatically. Basically, the pss analysis is re-run. Instead of starting from scratch, it starts from the solution that is in the *writepss* file. This is shown below.

```
/net/lnx-rdavis3/usr2/rdavis/demos/ic61/docs_database/simulation/D
File  Help                                                    cādence

    method = gear2only
    lteratio = 10
    relref = alllocal
    cmin = 0 F
    gmin = 1 pS
    rabsshort = 1 mOhm


Conv residual norm = 1.63.
Norm of periodicity discrepancy due to numerical error: 5.125692e-03.
Worst node: I0.I28.I9.ls11:1.
Conv DeltaSolution norm in fdtd = 165.
Conv residual and DeltaSolution norm in fdtd = 165.

Conv residual norm = 13.5e-03.
Norm of periodicity discrepancy due to numerical error: 3.548078e-06.
Worst node: I0.I28.I9.ls11:1.
Conv DeltaSolution norm in fdtd = 178.
Conv residual and DeltaSolution norm in fdtd = 178.

Conv residual norm = 1.03e-03.
Norm of periodicity discrepancy due to numerical error: 1.253602e-07.
Worst node: I0.I29.net023.
Conv DeltaSolution norm in fdtd = 403e-03.
Conv residual and DeltaSolution norm in fdtd = 403e-03.
Number of refinements using finite difference method = 1. Total steps

Finite difference refinement finished.

Number of accepted pss steps =              6955
Initial condition solution time: CPU = 0 s, elapsed = 0 s.
pss: The steady-state solution was achieved in 1 shooting-Newton itera
Number of accepted pss steps =              6955
Total time required for pss analysis `pss': CPU = 13.238 s, elapsed =
Time accumulated: CPU = 13.6779 s, elapsed = 20.0336 s.
Peak resident memory used = 193 Mbytes.

modelParameter: writing model parameter values to rawfile.
element: writing instance parameter values to rawfile.
outputParameter: writing output parameter values to rawfile.
designParamVals: writing netlist parameters to rawfile.
primitives: writing primitives to rawfile.
subckts: writing subcircuits to rawfile.

Aggregate audit (5:53:12 PM, Tue Aug 9, 2011):
Time used: CPU = 13.8 s, elapsed = 20.8 s, util. = 66.2%.
Time spent in licensing: elapsed = 488 ms.

39   Auto Update On
```


© 1994-2020 All Rights Reserved.

Checkpss (Shooting Only)

Leave this option at the default of *yes*. This option is used in conjunction with the *readpss* option and the *highorder* option. If *readpss* is set to a filename, (The pss is read from a file), and if *highorder* is set to *yes,* only then is *checkpss* read. If *checkpss* is set to *yes* (default), then any changes in the circuit are observed, and the *readpss* file serves as the starting point for the pss iterations. If the *checkpss* option is set to *no*, then none of the changes to the circuit elements are used. The readpss file is used directly with no changes.

Saveclock (Shooting and Harmonic Balance)

saveclock saves the state of the simulation in the tstab interval at the time interval in seconds specified by *saveclock*. When the clock time has passed in the tstab interval, the file specified by the option *savefile* is created. When subsequent clock time intervals have passed, the file is overwritten. Use only one of the save options at a time. The example below shows *saveclock* set to three minutes.

Saveperiod (Shooting and Harmonic Balance)

saveperiod saves the state of the simulation in the tstab interval at the simulation time interval in seconds specified by *saveperiod*. When the simulation time has passed in the tstab interval, the file specified by the option savefile is created. When subsequent simulation time intervals have passed, the file is overwritten. Use only one of the save options at a time. The example below shows *saveperiod* set to 25 nanoseconds.

Savetime (Shooting and Harmonic Balance)

savetime is a list of times in the tstab interval where the state of the simulation is written out. The list should be specified with spaces between the entries. The information is written out with the filename specified in the *savefile* option with extensions for the time after that. Use only one of the save options at a time.

The example below shows four times in the tstab interval to save.

© 1994-2020 All Rights Reserved.

Savefile (Shooting and Harmonic Balance)

This is the file name to write out the tstab state information to. If you do not specify a filename that starts with the slash (/) character, the file is stored in the *netlist* directory. To locate the *netlist* directory, first select *Setup - Simulator/Directory/Host* menu in ADE Explorer, and read the path to the project directory. This is the path to the *simulation* directory. In the simulation directory, navigate to the `<circuit_name>`*/spectre/<schematic or config>/ netlist* directory.

Note: Do not specify a relative path name like *./save_file/run1*.

Recover (Shooting and Harmonic Balance)

recover specifies the file name to recover the tstab simulation from. If *saveperiod* or *saveclock* are used to make the savefile, then just the same name as specified in *savefile* is used. If *savetimes* has a list of times specified, then several files are created at the times specified in the list. Click (...) and browse to the netlist directory where the savefiles are shown.

© 1994-2020 All Rights Reserved.

Select one from the list, and click *Open*. This adds the full path to the *recover* option, as shown below. Also note that in the example below, the *savefile* option is still set, and times after the restart times are specified. This is specifically allowed. As long as tstab is 200n or larger, the files at 175n and 200n will be added and can be reused later.

© 1994-2020 All Rights Reserved.

Now change tstab to a later time. In this case 200n was set.

© 1994-2020
All Rights Reserved.

When the simulation runs, the savefiles are created, as shown below.

© 1994-2020 All Rights Reserved.

The waveform from the original tstab and the restarted tstab overlay is shown below.

© 1994-2020
All Rights Reserved.

Even in the zoomed-in view, the waveforms overlay, as shown below.

© 1994-2020
All Rights Reserved.

Misc Tab

Step (Shooting Only)

Leave this option at the default. All the nodes are checked for curvature in the local truncation error (Numerical integration error) check. If that node has a capacitor or inductor connected to it, the timestep should be controlled to reduce the numerical integration error. If the node does not have a capacitor or inductor connected to it and it has a lot of curvature, the timestep can only be reduced to the value specified in the *step* option. The default is the simulation interval divided by 1000.

Maxstep (Shooting and Harmonic Balance)

In shooting, *maxstep* is observed in the tstab interval and in the shooting window. In harmonic balance, it is observed in the tstab interval only. Normally, *maxstep* should be left at the default and either setting *errpreset* or changing *reltol* is used to control the accuracy.

© 1994-2020 All Rights Reserved.

In some cases, extreme accuracy is needed, and this is accomplished by setting a small maxstep value. A small constant timestep produces maximum accuracy at the cost of runtime and memory consumption.

An example with *maxstep* set to 1 picosecond set is shown below.

Tstart (Shooting and Harmonic Balance)

Normally, this is not used. *tstart* specifies the starting time for the tstab interval and defaults to 0 (zero). A negative, 0, or positive *tstart* is allowed. An example with *tstart* set to 100n is shown below.

Use envelope for tstab (Shooting and Harmonic Balance)

This option is not recommended because we are considering ending support for this feature. Normally, a transient analysis is used for the tstab interval. In the case of circuits that have high frequencies and long time constants like oscillator amplitude control or crystal oscillators, the time required for the transient analysis can be excessive. In this case, select *yes* for *Use Envelope for tstab*, and set up a shooting envelope analysis. In the pss *Choosing Analyses* form, set tstab slightly larger than the stop time for the envelope analysis. The example below shows the *Use Envelope for tstab* option set to *yes*. For more information on the envelope analysis, see Chapter 6, “Envelope (ENVLP) Analysis”.

The envlp and pss *Choosing Analyses* forms are shown below.

© 1994-2020
All Rights Reserved.

© 1994-2020
All Rights Reserved.

© 1994-2020
All Rights Reserved.

memory usage estimation (Harmonic Balance only)

The *memory usage estimation* option provides an estimate of the memory required to perform Harmonic Balance analysis. If this option is set to *yes*, an estimate of the memory required to perform hb analysis is printed in the log file. You can use this estimate to plan the computing resources before performing hb analysis. In memory estimation mode, the simulator performs a short simulation and exits after printing the estimate in the log file without saving the harmonic balance results. You must set this option to *off* to perform actual hb analysis. Memory estimation is not recommended for simulations that require less than 500MB of disk space.

AdditionalParams (Shooting and Harmonic Balance)

This is a field for `<option_name>=value` statements. Multiple `<option_name>=value` statements are allowed with a space between them. Generally, this is used to unlock the beta features, however, there is one case where this field might be useful for an option that is not in the GUI.

In some cases, either in the tstab interval or in the shooting window, the timestep collapses to near zero. The symptom is that the percent complete stays at the same value for many reporting intervals. In this case, there might be a discontinuity in one of the models.

In transient simulation, two things can reduce the timestep; having too much numerical integration error or taking too many iterations at a single timestep can cause the timestep to be cut. Spectre does not report what is causing the timestep to become small. One way to eliminate the possibility of numerical integration error (which is called local truncation error in the simulator) as the cause of reducing the timestep is to set *lteratio* to 1e9. The disadvantage of this is that it disables the normal method of timestep control, therefore, *maxstep* must be set to maintain accuracy.

Another way to accomplish this is to set *lteminstep* to a value of about the stop time divided by 1e5. This allows the normal method of timestep control, but if for some reason the numerical integration error wants to cut the timestep too small, the error is ignored. *lteminstep* is the way to specify this. To do this in ADE Explorer, type `lteminstep=<Stop_Time>/1e5`. If you use `lteratio=1e9`, or if you set *lteminstep* and you still have very small timesteps, try increasing *maxiters* in the *Convergence* tab to between 40 and 100. When *lteratio* or *lteminstep* is set and you still have timestep problems, there is likely a discontinuity in either a device model or in a Verilog-A model. An example is shown below.

© 1994-2020 All Rights Reserved.

Help for Convergence Issues

Spectre can have difficulties in three areas:

- **Trouble with the time-zero timepoint**

  Occasionally, Spectre begins the simulation process, reports the inventory, and then reports that it is starting the tstab interval, and then appears to hang. If you have access to a CPU monitor, you will find that the CPU is at nearly 100% utilization. The likely cause is that the voltages from the signal sources at time=zero are non-zero, and the DC algorithm for calculating the time-zero timepoint is stuck either in `homotopy=source` or `gmin`. In this case, use the global options form (*Simulation* - *Options* - *Analog* in ADE Explorer) and set the *homotopy* option in the *Algorithm* tab to *dptran*. This is highly likely to resolve the problem. If this is ineffective, try setting *try_fast_op* to *no* in the global options *Algorithm* tab. If the problem persists, set `skipdc=sigrampup` and skip the time-zero calculation entirely.

- **Trouble in the tstab interval**

  The typical problem here is that in some cases, the timestep collapses to near zero, and as a result, the simulation appears to be stuck at a certain point. If this is the case, first try setting *relref* to *sigglobal*. This causes the accuracy to degrade a bit, but it allows the simulation to get past the troublesome point. If this fails, in the pss options form set the *lteminstep* option to about the tstab time divided by 1e5. If tstab is 100n, then type `lteminstep=100n/1e5` in the *additionalParams* field, and try again. If this does not fix the problem, there is likely a discontinuous model in the circuit. To be sure, set `maxiters=100, lteratio=1e9`, and `maxstep` about equal to `tstab/1e4`. Now run the simulation. If the problem persists, there is very likely a discontinuous model somewhere in the circuit. Use the Spectre output on which the nodes do not converge to determine which model has the problem.

  If you see the same problem in the shooting window, use the same series of steps.

- **Trouble in the shooting interval**

  Driven Circuits

  - Check that the option *iabstol* is not set smaller than 1e-12.
  - If there is a frequency divider in the circuit, remember that the input frequency will be divided down. Specify the divided-down frequency as the pss frequency. Remember that pss must simulate one cycle of the periodicity of the circuit in order to converge.
  - Set a longer tstab. A longer tstab allows the circuit to get closer to steady-state and thus converge better.

© 1994-2020 All Rights Reserved.

- Check the tstab stopping point. For a sine wave, you want to avoid having the tstab waveform stop at a peak or valley of the waveform. For a square wave, the best point to stop at in the tstab waveform is just before the next transition where most of the voltages have relatively settled.
- Try changing the tstab time by a fraction of a period. Just moving to a different point can help convergence. Generally, the longer the tstab, the higher the probability of convergence.
- Do not over-tighten *reltol* and *vabstol*. A practical minimum for *reltol* is 1e-6 and for *vabstol*, 1e-8. Although some circuits may allow smaller values, many would not.

Note: If the PSS analysis does not converge, the unconverged solution is not saved.

Oscillators

- Make sure that the oscillator is starting by selecting *yes* for *saveinit* and then viewing the startup waveform. If the waveform looks like a DC level for a while and then starts near the end of the tstab interval, try one of the methods listed in Different ways to start the oscillator on page 694 to start the oscillator more robustly.
- In some cases, Spectre will find a node that looks like it is a frequency divider output even though there is no divider in the circuit. When this happens frequently, convergence is negatively affected. Look at the starting and stopping times during the shooting iterations to see if this is happening and if you are sure that there is no frequency divider in the circuit, set the estimated frequency in the *Choosing Analyses* form to about 3.5 times the actual frequency of oscillation. The reason this works is that at the end of tstab, one period of the estimate is simulated, and then four periods of the estimated frequency is simulated, and during this interval, Spectre looks for a frequency divided output. Setting the estimated frequency to 3.5 times the real oscillation frequency causes just over one cycle of the real oscillation frequency to be simulated, and so Spectre cannot find a frequency-divided output.
- If there is a frequency divider in the circuit, make sure that the oscillator node is on the output of the frequency divider and the pss estimated frequency is the divided down frequency.
- Change tstab by a fraction of a period so that the starting and ending point are different. For sinusoidal oscillators, avoid peaks and valleys. For square wave oscillators, try to end tstab just before a transition. When using this strategy, note that the save and recover options can prevent the need to resimulate from time zero each time a different tstab is used.
- Try a longer tstab. Sometimes there are long time constants in the bias circuitry. When using this strategy, note that the *save* and *recover* options can prevent the need to resimulate from time zero each time a different tstab is used.

© 1994-2020 All Rights Reserved.

- Try setting `reltol=1e-5` and `vabstol=3e-8`. Setting tighter tolerances makes the amplitude and frequency estimate from the tstab interval more accurate.
- Try setting *maxacfreq* to about the frequency times 1000. This forces more timepoints, which makes the simulation more accurate. More time and more memory are required when this is set.
- If the *conv_norm* number is trending down with no large increases from iteration to iteration, try setting *maxperiods* to the 200 range. If *conv_norm* generally trends down during the additional iterations, set maxperiods to a higher value. If the *conv_norm* trends down, and then suddenly has a large increase, it is unlikely that setting *maxperiods* to a higher value will succeed.

Note: If the PSS analysis does not converge, the unconverged solution is not saved.

© 1994-2020 All Rights Reserved.

Examples

Generally speaking, this section shows examples using shooting. The reason is that harmonic balance has Chapter 3, which is dedicated to the hb *Choosing Analyses* form. All the capabilities shown are available for shooting and harmonic balance unless otherwise noted.

Pss Shooting with a single input to an amplifier or the LO applied in a mixer

The example below shows an amplifier.

© 1994-2020
All Rights Reserved.

1. In the schematic, for the RF port assign names for the sources in the *Frequency name 1* and *Frequency name 2* fields.

   This is just a name for you to use to remember the function of that signal. In this case, the name and the variable that sets the frequency are the same, but they do not need to agree. It is perfectly acceptable to have a name like RF1 and a frequency variable freq1. Additionally, the amplitude of both RF signals is set to the same variable name. This is not required either. In the case of measuring a large-signal IP3 where you deliberately want to match the amplitudes of both inputs, it is convenient to use the same name in both amplitudes.

© 1994-2020 All Rights Reserved.

In the case where you want the flexibility to introduce a blocking signal in addition to an RF signal, it makes sense to use two different variables because the blocker usually has a very high amplitude compared to the RF signal.

© 1994-2020
All Rights Reserved.

2. In ADE Explorer, assign values for the variables that are used in the input port. In this example, the first RF frequency is set to 2.4GHz, and the second RF frequency is set to 0, which disables that frequency from being generated.

3. In the ADE Explorer *Choosing Analyses* form, select pss in the *Analysis* section.

© 1994-2020 All Rights Reserved.

4. Select an engine that you want to use. *Shooting* and *Harmonic Balance* are both available in pss.

5. All the sources in the top-level circuit are read in to the list in the *Fundamental Tones* section. If you have signal sources lower in the level of the hierarchy, select *Update From Hierarchy*.

6. For the beat frequency, either select *Auto Calculate*, or type the frequency for the periodicity of the circuit.

If you are simulating a frequency divider, selecting *Auto Calculate* is inappropriate because it bases the calculation on the frequency of the source in the circuit. In this case, specify the frequency of the divider output.

7. Specify the number of harmonics that you want to calculate from the waveform in the *Number of harmonics* field.

If you do not want harmonics to be calculated, type 0 (zero) in the *Number of harmonics* field. When the number of harmonics is raised above 10, more timepoints are forced in the pss which raise the simulation time and the memory required for the simulation.

© 1994-2020 All Rights Reserved.

8. In most cases, moderate accuracy is enough. If you want more accuracy, select *conservative*. If this is not enough, refer to the section Accuracy and Settings and Trade-Offs on page 687.

9. Run the simulation.

© 1994-2020
All Rights Reserved.

10. When the simulation completes, select *Results - Direct Plot - Main Form* in ADE Explorer. The *Direct Plot Form* is displayed, as shown below.

© 1994-2020
All Rights Reserved.

Plotting the output spectrum in dBm

In the Direct Plot form:

1. Select *pss* from the *Analysis* section.
2. Select *Power* from the *Function* section.
3. To plot the power in dBm from a voltage on a net with an assumed resistor on that net, select *Net (specify R)* from the *Select* drop-down list.
4. Select *dBm* from the *Modifier* section.
5. Select the net in the schematic you want to plot.

The spectral plot is shown in the waveform tool.

© 1994-2020
All Rights Reserved.

Same Example Using Harmonic Balance

The hb *Choosing Analyses* form is only slightly different from the shooting form. The list of sources in the *Tones* section lists all the sources including the ones that have the frequency set to zero. There is an *Oversample Factor* field. There is also an hbhomotopy choice. For more information on harmonic balance, see Chapter 3, “Frequency Domain Analyses: Harmonic Balance”.

© 1994-2020 All Rights Reserved.

© 1994-2020
All Rights Reserved.

The waveform results agree for shooting and harmonic balance, as shown in the waveform tool.

Loadpull

Generally, loadpull is performed using harmonic balance. There are several reasons, but the biggest reason is that harmonic balance is almost always faster than shooting for a loadpull simulation. For harmonic balance loadpull information, see Loadpull on page 208.

© 1994-2020 All Rights Reserved.

PSS Shooting Loadpull

1. For a loadpull simulation, copy a a triplexer (for setting the reflection and angle for up to three harmonics) or the ten_plexer (for up to 10 harmonics) into a local writable library, and rename the cell. These components are located in the `rfLib` library.

© 1994-2020
All Rights Reserved.

2. Open the triplexer_local schematic, and select the portAdapter for the desired harmonic. The top portAdapter is for the first harmonic, and the harmonic numbers increase as you move down in the schematic. The top portAdapter is highlighted below.

© 1994-2020
All Rights Reserved.

The properties list is set as follows:

© 1994-2020
All Rights Reserved.

3. Change *Phase of Gamma (degrees)* and *Mag of Gamma (linear scale)* to variable names. Note that the phase is above the magnitude in the properties list.

© 1994-2020
All Rights Reserved.

4. Save the local copy.

© 1994-2020 All Rights Reserved.

5. Add the local copy of the triplexer or ten_plexer as the load in the schematic. Save the schematic.

© 1994-2020
All Rights Reserved.

© 1994-2020
All Rights Reserved.

6. Add the two variables you used in the local triplexer or ten_plexer into ADE Explorer.

The example above shows `mag` and `theta` variables added. These variables should not be used to set any component values in the circuit.

Triplexer from your local library directory contains three filters for the first, second, and third harmonics and portAdapters with port loads for each of the three filters.

© 1994-2020 All Rights Reserved.

7. Open the PSS *Choose Analyses* form.

© 1994-2020
All Rights Reserved.

8. Set up the top of the form as usual.
9. Select the *Loadpull* checkbox.

10. Click *Select* on the right of the greyed out *Load Instance* field.
11. In the triplexer_local schematic, select the topmost portAdapter on the right side of the schematic if you want a loadpull for the first harmonic. Select the next portAdapter down for a loadpull of the second harmonic. Select the third portAdapter down for the third harmonic. The first harmonic portAdapter is highlighted below.

© 1994-2020
All Rights Reserved.

The instance name is entered in the *Load Instance* field. Direct entry by typing is not allowed for this field.

12. Click the *Variable* button on the right of the *rho (Z)* field. Select the variable you entered in ADE Explorer.

This variable is used to set the value of the reflection coefficient in the port.

13. Specify a sweep range, and set the spacing.

14. When completed, select *OK*.

The values are displayed in the *Choose Analyses* form.

15. Click the *Variable* button on the right of the *phi (Z)* field. This sets the angle of the reflection coefficient.

© 1994-2020
All Rights Reserved.

16. In the *Select Design Variable* window, select the second variable you entered and specify the sweep limits.

17. When completed, click *OK*.

    The variables are updated in the *Choose Analyses* form.

18. When completed, click *OK* and run the simulation in ADE Explorer.
19. In ADE Explorer, select *Results - Direct Plot - Main Form*.

There are two methods of plotting loadpull curves.

First method

1. Select *pss* from the *Analysis* section.
2. Select *Power Contours* from the *Function* section.
3. Select *Magnitude*, *dB10*, or *dBm* as the power modifier.

© 1994-2020
All Rights Reserved.

4. Specify 2 more than the number of contour lines you want.

   The smallest and largest power occurs at a single point which is plotted, but are very difficult to see in the waveform tool.

5. Select the *Close Contours* check box. This causes the contours from the loadpull to be displayed as circles or ellipses, depending on the distortion of the circuit.

6. Select the first harmonic.

7. Select the top terminal of the top port in the triplexer.

   The power contours appear.

© 1994-2020
All Rights Reserved.

Second method

1. Select *pss results*.
2. Select *Power* from the *Function* section.
3. Select *variable* from the *Sweep* section.
4. Select *Magnitude*, *dB10*, or *dBm* from the *Modifier* section.
5. Select the *harmonic number*.
6. Select *Loadpull Contour*.
7. Specify 2 more than the number of curves you want.
8. Select the *Closed Contours* check box. This causes the contours from the loadpull to be displayed as circles or ellipses, depending on the distortion of the circuit.
9. Select the top terminal of the top port of the triplexer circuit.

© 1994-2020 All Rights Reserved.

The loadpull curves appear in the waveform tool.

Two Input Frequencies

Two input frequencies can be simulated as long as the beat frequency is high compared to the frequencies that are applied to the circuit. In general, if the pss beat frequency is smaller than about 1/25th the highest input frequency, the simulation is faster if you use qpss or hb.

In the example below, 30 periods of 900MHz and 31 periods of 960 MHz need to be calculated. This is on the edge of acceptability for pss. For this simulation, it might be faster

© 1994-2020 All Rights Reserved.

to simulate using qpss or hb because fewer harmonics need to be solved, which requires less memory and time.

© 1994-2020
All Rights Reserved.

Note that when the frequencies are moved slightly to 910M and 970M, the beat frequency drops to 10MHz. In this case, it would definitely be faster to use hb or qpss. These analyses are much faster than pss. Also, since the pss frequency is lower, more harmonics need to be calculated in order to see the desired mixing products.

© 1994-2020
All Rights Reserved.

Sweeps

For the next plots, a sweep is set up in the pss *Choosing Analyses* form. A sweep is needed in order to measure compression.

Sweeps can be made either in the parametric plot tool in ADE Explorer where the number of swept things is not limited, or in the pss *Choosing Analyses* form, where the number of swept things is limited to three. This example shows using the pss *Choosing Analyses* form shown below. For documentation on the parametric plot tool, please refer to the ADE Explorer documentation.

© 1994-2020 All Rights Reserved.

© 1994-2020
All Rights Reserved.

1. Select a sweep number from the *Sweep* drop-down list.

2. Select the type of parameter that is being varied from the *Variable* drop-down list.

3. Type the variable name in the *Variable Name* field. You can also select the variable from a list. To do this, click *Select Design Variable*. and the list is displayed.

Select the desired variable, and click *OK* to enter the variable.

© 1994-2020
All Rights Reserved.

4. Define the sweep range. Set up a sweep range, and if you desire values that are in between the sweep values, or above or below the sweep values, select *Add Specific Points*. Type a list of values separated by spaces. This is shown below.

For many power measurements, currents need to be saved. If you are using Shooting, these currents can be saved, To save all the currents from the ADE Explorer environment. Select *Outputs* - *Save All*, and then select *all* for *Select Device Currents (currents)*. This takes

© 1994-2020 All Rights Reserved.

the currents that are calculated from every component in the circuit and adds them into the simulator result file.

For hb, port, and voltage source, only currents are saved, and this save is automatic. The ADE Explorer *Save Options* form shown above has no effect in hb. If you need to measure a current other than in a port or voltage source, add an iprobe in series with the branch.

© 1994-2020 All Rights Reserved.

If all the currents need to be saved when harmonic balance is used as the engine, select *yes* for the *Select AC terminal currents (useprobes)* option above. This adds a zero volt battery in series with every branch in the circuit. Although this results in saving all the currents, it also adds entries in the harmonic balance matrices for each voltage source. As a result, the hb matrices get much larger, therefore, more time and memory are required to solve the equations.

In addition, there is a very small chance of causing unresolvable numerical issues and the simulator issues a singular matrix error message. If you see this error, turn off the *Select AC terminal currents (useprobes)* option and run the simulation again.

Note: The *Select AC terminal currents (useprobes)* option is not recommended if you run a small-signal analysis after the large-signal shooting pss analysis.

Compression Point

This requires a swept input power setup as shown in the section Loadpull on page 762.

Make sure that the power at the first point in the sweep is low enough so that there is no compression in the output at that power level. The first point in the sweep is used to establish the uncompressed power for the compression measurement.

To plot a compression point measurement:

1. Select *Results - Direct Plot - Main Form* in ADE Explorer.

© 1994-2020 All Rights Reserved.

The Direct Plot Form is displayed, as shown below.

2. Select *Compression Point* from the *Function* section.

3. Select the type of thing that needs to be probed in the circuit from the *Select* drop-down list.

© 1994-2020
All Rights Reserved.

4. Specify the compression level.
5. Specify input or output-referred compression amount.
6. Select the output harmonic to be plotted.
7. Make the selection that is shown at the bottom of the *Direct Plot Form*. In this example, a port in the schematic that is used as a load should be selected.

   The compression curve is displayed.

Voltage Gain

This example shows a swept measurement. To set this up, see Loadpull on page 762. A sweep is not required for this measurement.

The voltage gain direct plot function provides the output spectrum divided by the amplitude of the input at the specified input frequency. Note that the input amplitude at the input

© 1994-2020 All Rights Reserved.

frequency has a single value. All the output harmonics at all the calculated output frequencies are scaled by the single voltage at the input.

To plot the voltage gain:

1. Select *Voltage Gain* from *Function* section.
2. Select the appropriate modifier.

© 1994-2020
All Rights Reserved.

3. Select the power level of the sweep.
4. Select the input harmonic frequency.
5. Select the output node in the schematic.
6. Select the input node in the schematic.

Below is the output spectrum of the circuit, divided by the amplitude of the input at the input frequency.

Power Gain

This example shows a swept measurement. To set this up, see Loadpull on page 762. A sweep is not required for this measurement.

The power gain direct plot function provides the output spectral power divided by the power of the input at the specified input frequency. Note that the input power at the input frequency has a single value. All the output harmonics at all the calculated output frequencies are scaled by the single voltage at the input.

© 1994-2020 All Rights Reserved.

To plot the power gain of a circuit:

© 1994-2020
All Rights Reserved.

1. Select *Power Gain* from the *Function* section.
2. Select the appropriate modifier.
3. Select the input power.
4. Select the frequency of the input.
5. Select the terminal of the load.
6. Select the terminal of the input source.

Below is the output power spectrum, divided by the magnitude of the input power at the input frequency.

© 1994-2020
All Rights Reserved.

Power-Added Efficiency

This example shows a swept measurement. To set this up, see Loadpull on page 762. A sweep is not required for this measurement.

To plot power-added efficiency:

© 1994-2020
All Rights Reserved.

1. Select *Power Added Eff* from the *Function* section.
2. Select the frequency of the input.
3. Select the number of DC sources that power the circuit.
4. Select the terminal of the output load.

   > Select Numerator Output Instance Terminal on schematic..

5. Select the terminal of the input source.

   Select Numerator Input Instance Terminal on schematic...

6. Select the terminal(s) of the DC sources.

   > Select # 1 Denominator DC Instance Terminal on schematic...

The swept PAE curve is displayed.

© 1994-2020
All Rights Reserved.

Compression Versus Output Power

This example shows a swept measurement. To set this up, see Loadpull on page 762.

To plot compression versus output power:

© 1994-2020
All Rights Reserved.

1. Select *Comp Vs Pout* from the *Function* section.
2. Select the output frequency.
3. Select the load port in the circuit.

The compression curve is displayed. The assumption is that the first point in the sweep has no compression.

Power Gain Versus Output Power

This example shows a swept measurement. To set this up, see Loadpull on page 762.

© 1994-2020
All Rights Reserved.

To see the power gain versus the output power:

© 1994-2020 All Rights Reserved.

1. Select *Power Gain vs Pout* from the *Function* section.
2. Select the appropriate modifier.
3. Select the output frequency.
4. Select the input frequency.
5. Select the terminal of the load.
6. Select the terminal of the input source.

The plot is displayed. The vertical axis is the power gain. The horizontal axis is the swept input power.

Plotting THD

This example shows a swept measurement. To set this up, see Loadpull on page 762. A sweep is not required for this measurement.

© 1994-2020
All Rights Reserved.

To plot the total harmonic distortion:

1. Select *THD* from the *Function* section.
2. Select the output frequency.
3. Select the net in the schematic.

© 1994-2020 All Rights Reserved.

The THD versus input power plot is displayed.

Large-signal IP3

This example shows a swept measurement. To set this up, see Loadpull on page 762.

In pss, all the harmonics of the input waveform are calculated. If the input frequencies get close together, the pss beat frequency becomes small compared to the highest input frequency. If you need to simulate more than 25 periods of the highest input frequency, switch to hb or qpss which run faster than pss.

To plot a large-signal IP3, both inputs are enabled in the ADE Explorer window by specifying the values under the *Design Variables* section of the ADE Explorer for the frequency and

© 1994-2020 All Rights Reserved.

amplitude of both RF sources. Note that in this case, the highest frequency (950MHz) divided by the beat frequency (50MHz) is 19. This is a reasonable problem for pss.

The input is swept over a reasonable range of power levels. If the lower boundary of the sweep is too small, the noise floor of the simulator will be present in the result. If the input power gets too high, the inter-modulation product gets unpredictable. In this example, the

© 1994-2020 All Rights Reserved.

power is deliberately started at a very small value and the stopping power is in a very nonlinear region. This is done so the proper way to plot an IP3 can be shown later,

© 1994-2020
All Rights Reserved.

1. When the setup is complete, run the simulation.
2. When the simulation completes, select *Results - Direct Plot - Main Form* in ADE Explorer.

© 1994-2020
All Rights Reserved.

3. Select *IPN Curves* from the *Function* section.
4. Select the type of output signal to plot (or Port) The default is to select a port in the schematic.
5. Select *Variable Sweep*.
6. Set the order of the curves to be plotted. Usually, the order is 2nd or 3rd.
7. Select whether you want input or output-referred IP3 or IP2.
8. Leave the extrapolation point blank for the first plot (explained later).
9. Select the third or second-order frequency from the 2nd or *3rd Order Harmonic* field.

   Usually, you select the frequency that has the highest amplitude, which is often the lower frequency term.
10. Select the first-order frequency from the *1st Order Harmonic* field. Usually you want the smallest amplitude, which is often the higher frequency term.
11. Select the output port in the schematic.

© 1994-2020 All Rights Reserved.

The swept large-signal IP3 is plotted, as shown below.

- Plotting IP3 always involves setting an extrapolation point. First plot the curve using the default for the extrapolation point.
- At high power, the third-order curve becomes unpredictable. At low power, the simulation has a noise floor. The data from the intermodulation product starts to level off.
- Set the extrapolation point above the noise floor, but below the large-signal region.
- The actual data for the intermodulation product should follow the ideal curve for at least one data point in the sweep above and below the extrapolation point.

© 1994-2020
All Rights Reserved.

- When -30 is set as the extrapolation point, the data for the third-order frequency follows the ideal curve above and below the extrapolation point, as shown below.

© 1994-2020
All Rights Reserved.

- Note that the IP3 calculation has increased by about 7dB. Make sure that the extrapolation point is set correctly to get the correct measurement.

- The above plot is actually a small-signal IP3 measurement. If this is your objective, using rapid IP3 or IP2 in the AC *Choosing Analyses* form for amplifiers and in the PAC *Choosing Analyses* form for circuits that translate frequencies will run much faster and will provide a solution that is just as accurate.

- Note that `bsim3` and `bsim4` models have a limitation that causes the IP3 measurement to become inaccurate for switched-FET mixers whether it is done in rapid IP3, pss, qpss, or hb. In this case, the psp model is highly recommended. If this model is not available from the foundry, use pss, qpss, or hb at as high an input amplitude as possible without causing large-signal behavior in the third-order product.

© 1994-2020 All Rights Reserved.

Oscillator Simulation

There are no input signals for an oscillator. In the pss analysis, the waveform needs to be calculated. To do this, both the frequency and the amplitudes need to be solved. This requires more iterations to calculate the solution.

In ADE Explorer, variables are used to set the tuning voltage and the power supply voltage.

Setting Dynamic Parameters in Tstab

To set the dynamic parameters:

© 1994-2020
All Rights Reserved.

1. Open the PSS *Choosing Analyses* form.

© 1994-2020 All Rights Reserved.

2. Specify an estimate of the oscillation frequency and set the number of harmonics to be calculated.
3. Select *conservative*.
4. Choose *Yes* for *Run transient?*.
5. Specify the stop time for tstab in the *Stop Time (tstab)* field.
6. Select *yes* for *Save Initial Transient Results (saveinit)*.
7. Select *Dynamic Parameter*.
8. Choose *Parameter vector* from the drop-down list.
9. Specify the name of the option that you want to vary in the tstab interval in the *Parameter name* field. The example shows *reltol*.
10. Specify 0 (zero) in the *Time* field.
11. Specify a reasonable value for the option. This example steps reltol from 1e-2 to 1e-7 every 40 nsec in the tstab interval. Note that 1e-2 is a very large value for reltol.
12. Click Add.
13. Add other time-value pairs as appropriate.
14. Select *Oscillator*.
15. Click Select to the right of the *Oscillator node+* field.
16. Select a net in the schematic that has an oscillator signal.
17. Click *OK*.

© 1994-2020 All Rights Reserved.

18. Run the simulation. The setting for the option appear in the Spectre output window.

```
/servers/scratch03/kamal/spectrerftestcase151/RF_Do
File  Edit  View  Help                                   cadence

    pss: time = 6.33 ns      (2.5 %), step = 21.05 ps    (8.33 m%)
    pss: time = 18.97 ns     (7.5 %), step = 10.53 ps    (4.16 m%)
    pss: time = 31.61 ns    (12.5 %), step = 21.05 ps    (8.33 m%)

Notice from spectre at time = 40 ns during periodic steady state analys
    The value of parameter 'reltol' was set to 0.001 at time 4e-08 s.

    pss: time = 44.24 ns    (17.5 %), step = 10.15 ps    (4.02 m%)
    pss: time = 56.9 ns     (22.5 %), step = 19.34 ps    (7.65 m%)
    pss: time = 69.53 ns    (27.5 %), step = 5.263 ps    (2.08 m%)

Notice from spectre at time = 80 ns during periodic steady state analys
    The value of parameter 'reltol' was set to 0.0001 at time 8e-08 s.

    pss: time = 82.17 ns    (32.5 %), step = 2.526 ps     (999 u%)
    pss: time = 94.81 ns    (37.5 %), step = 11.67 ps    (4.62 m%)
    pss: time = 107.5 ns    (42.5 %), step = 9.455 ps    (3.74 m%)

Notice from spectre at time = 120 ns during periodic steady state analys
    The value of parameter 'reltol' was set to 1e-05 at time 1.2e-07 s.

    pss: time = 120.1 ns    (47.5 %), step = 1.83 ps      (724 u%)
    pss: time = 132.7 ns    (52.5 %), step = 4.392 ps    (1.74 m%)
    pss: time = 145.4 ns    (57.5 %), step = 4.185 ps    (1.66 m%)
    pss: time = 158 ns      (62.5 %), step = 1.833 ps     (725 u%)

Notice from spectre at time = 160 ns during periodic steady state analys
    The value of parameter 'reltol' was set to 1e-06 at time 1.6e-07 s.
        Further occurrences of this notice will be suppressed.

    pss: time = 170.7 ns    (67.5 %), step = 2.425 ps     (959 u%)
    pss: time = 183.3 ns    (72.5 %), step = 2.074 ps     (820 u%)
    pss: time = 195.9 ns    (77.5 %), step = 836.3 fs     (331 u%)
    pss: time = 208.6 ns    (82.5 %), step = 1.073 ps     (424 u%)
    pss: time = 221.2 ns    (87.5 %), step = 970 fs       (384 u%)
    pss: time = 233.9 ns    (92.5 %), step = 354.5 fs     (140 u%)
    pss: time = 246.5 ns    (97.5 %), step = 1.008 ps     (399 u%)
The Estimated oscillating frequency from Tstab Tran is = 1.83319 GHz .

Notice from spectre during periodic steady state analysis `pss'.
    The value of parameter 'reltol' was set to 1e-07 at time 2.52821e-07
9  >                                                     L1      C1
```


© 1994-2020 All Rights Reserved.

19. When the simulation finishes, select *Results - Direct Plot - Main Form*.

20. Select *tstab* from the *Analysis* section.

21. Click a net in the schematic with the oscillator signal on it. A waveform is plotted.

© 1994-2020
All Rights Reserved.

22. To measure the instantaneous frequency, select the waveform and choose *Measurements - Derived Plots*.

© 1994-2020
All Rights Reserved.

23. In the *Derived plots* window, deselect the *Period* and *Dutycycle* options.

24. Click *OK*.

© 1994-2020
All Rights Reserved.

25. The instantaneous frequency is plotted.

Note the effect of reltol on the frequency. When reltol is 1e-2, the frequency is low and not very stable. When reltol is 1e-3, the frequency increases and is slightly more stable. As reltol gets smaller, the frequency creeps up and values smaller than 1e-4 produce stable frequencies. Setting values smaller than 1e-5 do not have much effect on stability or frequency, but do have the effect of increasing simulation time.

Normal Oscillator Simulation

1. In the pss *Choosing Analyses* form:
2. Select *pss*.

© 1994-2020
All Rights Reserved.

3. Select the engine that you want to use.

4. Specify a frequency estimate that is between 0.5 and 1.5 times the actual oscillation frequency if you are using the linear oscillator initial condition, or from 0.25 to four times the actual oscillating frequency if you use the default oscic.

5. Set the number of harmonics you want for the spectral calculation.

6. Select the accuracy. *conservative* accuracy is recommended for oscillators.

7. Set *Run transient?* to *Decide automatically*. This automates the setting of tstab and switches to the pss algorithm as soon as steady-state is reached.

8. If you want to see the startup waveform, select yes for *saveinit*.

9. Select *Oscillator*. If you have a single-ended oscillator, specify one node inside the feedback system for the *Oscillator node*. If you have a differential oscillator, specify two nodes at mirror locations in the *Oscillator node+* and the *Oscillator node-*.

© 1994-2020
All Rights Reserved.

10. Select the *Calculate initial conditions (ic) automatically* option.

11. Now run the simulation. When the simulation completes, select *Results - Direct Plot - Main form* in ADE Explorer.

Plotting the Oscillator Startup Waveform

To plot the startup waveform, select tstab results and then select a node in the circuit. If you do not see the tstab results, go back to the *Choosing Analyses* form and select *yes* for *saveinit*.

For linear oscic, the startup characteristics of the waveform are not very evident. If you want to see the startup waveform to evaluate how robustly your oscillator starts up, use the default oscic and provide a very small starting kick at time zero plus so that the waveform starts small in amplitude and builds up. In this case, it is likely that tstab needs to be longer to allow the oscillator to reach a near steady-state operation before going into the pss solver.

The result is displayed in the waveform tool. Note that with the linear oscic, the oscillator starts at time = zero with the estimate of the frequency and amplitude from the very beginning. This

© 1994-2020 All Rights Reserved.

allows a very short tstab. Also note that the simulation continues after the setting of tstab. During this time, Spectre searches the oscillator for a divided-down output frequency.

© 1994-2020
All Rights Reserved.

Plotting the Oscillator Output Spectrum

To plot the output spectrum, select *pss*, and then select either *Voltage* or *Power*.

© 1994-2020
All Rights Reserved.

When a node is selected in the schematic, the power in dBm is plotted in the waveform tool, as shown below.

Additions for Oscillator Swept Tuning Voltage

In the pss *Choosing Analyses* form:

1. Select *Sweep*.
2. Type the variable name that sets the tuning voltage in the *Variable Name* field.

© 1994-2020
All Rights Reserved.

3. Set a reasonable sweep range for your oscillator.

© 1994-2020
All Rights Reserved.

4. Run the simulation.

5. When the simulation completes, select Results - *Direct Plot* - *Main Form* in ADE Explorer. The *Direct Plot Form* is displayed.

© 1994-2020
All Rights Reserved.

Plotting Output Power Versus Tuning Voltage

To plot the power as a function of the tuning voltage:

1. Select *pss* from the *Analysis* section.
2. Select *Power* from the *Function* section.
3. Select *variable* from the *Sweep* section.
4. Select *dBm* from the *Modifier* section.
5. Select the harmonic number. The frequency range is also shown.
6. Select the node in the schematic you want to plot.

The output power versus the tuning voltage is plotted, as shown below.

© 1994-2020
All Rights Reserved.

Plotting Output Power Versus Output Frequency

Now plot the harmonic frequency on the same plot.

© 1994-2020
All Rights Reserved.

The frequency is plotted on the same subwindow. A second vertical axis is used.

To plot the output power versus the frequency, right click on one of the numbers on the X Axis, and select Y vs. Y. The *Y vs Y* window appears. Select the frequency for the X Axis from the *Select a Trace* drop-down list and click *OK*.

© 1994-2020
All Rights Reserved.

The plot of the output power versus the output frequency is plotted in a new subwindow.

© 1994-2020
All Rights Reserved.

Plotting the Output Frequency Versus the Tuning Voltage

Now quit the waveform window, and select *Add To Outputs* in the *Direct Plot Form*, and plot the harmonic frequency again.

© 1994-2020
All Rights Reserved.

The tuning curve is plotted again, as shown below.

© 1994-2020
All Rights Reserved.

In ADE Explorer, a new expression is added under the maestro tab.

© 1994-2020 All Rights Reserved.

Plotting the Modulation Sensitivity

1. Double-click the expression in the *Details* column under the *maestro* tab.
2. Click the (...) button.

   The calculator opens with the expression in it.

© 1994-2020 All Rights Reserved.

3. Select the derivative function. The expression changes.

4. Set the plotting mode to *New Subwindow*.

5. Click the *Plot* icon.

© 1994-2020 All Rights Reserved.

The modulation sensitivity in MHz/Volt is plotted in the subwindow on the right.

Oscillator Tuning Mode Analysis

The oscillator tuning mode analysis tunes the oscillator so that phase noise can be measured at the desired oscillation frequency. It is supported in both harmonic balance and shooting methods and is typically used in Monte Carlo simulations or sweeps to obtain the phase noise.

In oscillator tuning mode analysis, the oscillator frequency is treated as the target frequency, and the value of a circuit parameter is obtained to achieve the target frequency. The circuit parameter can be an arbitrary ADE variable or an instance parameter.

1. Open the pss *Choosing Analyses* form.
2. Set up the target frequency in the *Beat Frequency* field.
3. Select the *Oscillator* check box.
4. Specify the output node from the `Oscillator node+` field. Alternatively, click *Select* and choose the output node from the schematic. Here, `vc2` is selected.

© 1994-2020
All Rights Reserved.

5. Select the *Enable tuning mode analysis* check box.

6. Choose the parameter that needs to be tuned from the drop-down list. Here, *Component Param.* has been selected.

7. Specify the name of the parameter in the *Component Name* field. Alternatively, click *Select Component* and select the parameter from the schematic. When you select a

© 1994-2020
All Rights Reserved.

parameter from the schematic a pop-up window appears that contains a list of selectable parameters.

8. Select a parameter from the list. The *Parameter Name* field is automatically populated with the name of the parameter (*dc* in the figure above).
9. Select the *Tuning range* check box and specify the minimum and maximum ranges for the parameter, if required.
10. Run the simulation.

© 1994-2020 All Rights Reserved.

11. After the simulation completes, click *Results - Direct Plot - Main Form* to display the *Direct Plot Form*.

12. Select *Tuning Parameter* from the *Function* section.

13. Click *Add to Outputs*.

14. Click *OK*.

© 1994-2020
All Rights Reserved.

The value of the tuning parameter is added in the *Value* column under the *maestro* tab of the ADE Explorer window, as shown below.

Oscillator Tuning Mode With Sweeps

Sweeps and Monte Carlo are also supported with tuning mode. Below is an example of a sweep of temperature with tuning mode and a phase noise measurement.

© 1994-2020
All Rights Reserved.

1. Enable sweeps in the pss *Choosing Analyses* form. In this example, temperature is swept from -20 t0 120C. Tuning mode is enabled as before.

© 1994-2020
All Rights Reserved.

2. Because of the sweep, pnoise has been changed to a single frequency offset, which simplifies plotting of phase noise after the sweep.

© 1994-2020
All Rights Reserved.

3. Now run the simulation. Because the tuning parameter and phase noise have already been added under the *Maestro* tab in ADE Explorer in the previous section, the tuning voltage and the phase noise are plotted after the simulation.

© 1994-2020
All Rights Reserved.

4. To make the phase noise plot more useful, select the X Axis, click the right mouse button, and select *Swap Sweep Var*. In the window that pops up, click *OK*. Then delete the original phase noise plot.

5. The trend in tuning voltage and phase noise is plotted.

© 1994-2020
All Rights Reserved.

Periodic AC Analysis (PAC)

Small-signal Versus Large-Signal Analysis for Conversion Gain

One way to measure conversion gain is to apply the LO and the RF signals in harmonic balance (hb) or quasi-periodic steady-state (qpss) and calculate the full large-signal response at each RF input frequency, and then plot the output as a function of the sweep frequency. While this works, because all the large-signal effects and many harmonics are calculated, the required time for the simulation can be significant.

The alternative is to use a small-signal approach similar to the AC analysis. The AC analysis runs considerably faster than running the transient at multiple frequencies because it ignores the large-signal effects. In circuits with frequency translation, the AC analysis is unusable because the calculation is based on the DC bias point. In hb Periodic AC (PAC) analysis, instead of using the DC bias point, the harmonic balance periodic steady-state (pss) solution with just the LO is used as the basis of the calculation. The large-signal analysis only calculates the harmonics of a single input, and no mixing products, and then hb pac is run after the hb pss analysis and calculates the outputs that are produced by mixing the pac input with the harmonics that are in the pss analysis. The overall time savings can be very significant.

© 1994-2020 All Rights Reserved.

Example: Conversion Gain (Down Conversion)

Consider the double-balanced diode mixer shown below.

The RF input source has the frequency and amplitude set to variable names in order to allow changing the test setup without needing to change the circuit. This is suggested for all the sources used in the circuit that might need to be changed. Also note that the pac magnitude term is set to 1 volt. This makes it easy for a conversion gain measurement because the input signal is 0dB when the *dB20* modifier is used in the *Direct Plot Form* for the pac result.

© 1994-2020 All Rights Reserved.

The property list is shown below.

© 1994-2020
All Rights Reserved.

The easiest way to get the variables list into ADE Explorer is to select *Variables - Copy From Cellview*. To set the values, select the value field to the right of the variable, type the desired value, and press *Enter*.

The LO frequency is 1GHz and its amplitude is 1.4 volts peak. This is +13 dBm. (+13 dBm is a common drive level for diode mixers.) The RF frequency is 1.1GHz, but it is disabled because the amplitude is set to 0 (zero). Setting either the frequency or the amplitude to zero disables the production of that waveform from the port. This provides an easy way to enable or disable signals without needing to change the schematic.

The pss analysis needs to be set up first. Note that both shooting and harmonic balance are available in pss for the engine. The example below shows a single input at 1GHz with 10 harmonics using the shooting engine. For more information, see Overview of Periodic Steady-State (pss) Analysis on page 647. Diode mixers have rapid changes in the diode

© 1994-2020 All Rights Reserved.

currents. Because of this rapid change, shooting is likely to be faster and more accurate than harmonic balance.

© 1994-2020
All Rights Reserved.

Next, the PAC analysis needs to be set up. Like AC, the frequency range specified in the form is the input frequency range. When shooting is used for the pss engine, *Maximum sideband* should be set somewhere less than four times the number of harmonics set in the pss analysis. Setting 10 sidebands calculates the mixing products from the input frequency in PAC with the harmonics through the tenth harmonic in the pss analysis. The actual number chosen depends on the mixing products that are important in your application. When harmonic balance is chosen for the engine, leave the *Maximum sideband* field blank.

© 1994-2020
All Rights Reserved.

Once the analyses are set up, the simulation can be run by selecting *Simulation - Netlist and Run* in ADE Explorer. When the simulation finishes, select *Results - Direct Plot - Main Form*.

The example below shows a typical *Direct Plot Form* setup to measure the conversion gain.

When the net in the schematic is selected, the waveform is displayed.

Note that all the short line segments are the mixing products that are produced when mixed with the harmonics in the periodic steady-state (pss) large-signal analysis.

© 1994-2020
All Rights Reserved.

In the down-conversion case, the frequencies that are produced for the down-converted output product are desired. In the *Direct Plot Form*, select *sideband* instead of *spectrum*,

© 1994-2020 All Rights Reserved.

and then select the desired frequency range from the list at the bottom of the form. Only that mixing product will be displayed, as shown below.

The conversion loss is about 6dB.

© 1994-2020
All Rights Reserved.

Example: Conversion Gain (Up Conversion)

The process is similar for an up-conversion mixer. In this case, the input frequency is low, as shown below. The LO remains at 1GHz, as it was in the previous section.

© 1994-2020
All Rights Reserved.

The RF outputs are 50MHz through 150MHz above and below the 1 GHz LO.

© 1994-2020
All Rights Reserved.

In the *Direct Plot Form*, both the outputs above and below the LO frequency might be desired.

Select the first frequency, and click the second while holding the *Ctrl* key. Now select the output net in the circuit and the results will plot.

© 1994-2020
All Rights Reserved.

Note that the conversion loss is almost identical for both mixing products.

PAC General Principles

The pss solution with LO calculates the amplitude of the harmonics and characterizes the nonlinearity of the system. Therefore, in pac, the output mixing products of the input signal mixing with the LO harmonics can be calculated. Like ac analysis, pac allows the input to be swept in frequency to calculate a frequency response of the output mixing products. Also, like ac, when pac is used for conversion gain measurements, the conversion gain is the same regardless of the input amplitude. The small-signal conversion gain is calculated and is a good representation of the gain until compression is approached or exceeded in the real circuit. If a measurement of compression is required, use a large-signal analysis like pss, hb, or qpss.

© 1994-2020
All Rights Reserved.

PAC used with Harmonic Balance PSS

When harmonic balance is used for the pss analysis, the harmonics of the circuit are calculated directly by the harmonic balance engine.

Think of a perfectly linear circuit with a single input applied. Because the circuit is perfectly linear, in the pss analysis when harmonic balance is used, the frequency domain solution only has the harmonic at the input frequency at every net in the circuit. No other harmonics are created because the system is perfectly linear. Because the circuit is linear, if a second input is applied, no frequency translation can occur.

Now imagine a real circuit with a single small input amplitude signal at 1GHz. In the frequency domain with harmonic balance selected in pss, the first harmonic is present along with the higher harmonics. Because the input signal is small, the signals at the harmonics of 1GHz are very small. If a second input is applied at 1.1 GHz, that signal is weakly converted when it is mixed with the harmonics of the circuit. Certainly, 1.1GHz would appear at the output with no mixing. 100MHz and 2.1GHz are produced by mixing with the 1GHz fundamental frequency. 900MHz and 3.1GHz are produced when 1.1GHz mixes with the second harmonic of the 1GHz input. 1.1GHz mixes with the other harmonics as well, producing a wide range of output frequencies from a single input frequency. The harmonic output might look like the figure below. Only the first three harmonics are shown. The black harmonics are the pss harmonics, and the red mixing products are pac outputs, which are called sidebands.

© 1994-2020 All Rights Reserved.

Note that there is an output mixing product that is created at 100MHz. Because the real circuit is nonlinear, harmonics of 100MHz are created in the circuit. In pac, the assumption is that all the signals are small, and with this assumption, the harmonics of 100MHz do not need to be calculated. This assumption also applies to all the other outputs in pac. Therefore, the PAC output is a partial output from the circuit with only the main mixing products displayed. Since in most cases, these are the desired outputs, conversion gain can still be obtained by using pac. In other words, pac calculates the main mixing products that are created by mixing with the pss harmonics, but the harmonics of the pac outputs are not calculated.

Now imagine that the 1GHz signal is larger. If the input goes up 10dB, the first harmonic goes up 10dB, In the frequency domain, if harmonic balance is selected in pss, the second harmonic would come up 20dB, the third harmonic would come up 30dB, and so on. If a second input at 1.1GHz is applied to the circuit, this mixes with the fundamental and the harmonics of 1GHz and more strongly mixes up-and-down in frequency. This is illustrated below. The black harmonics are the pss harmonics, and the red mixing products are pac outputs, which are called sidebands.

© 1994-2020
All Rights Reserved.

PAC used with Shooting PSS

Shooting is a variation of the transient analysis where the individual timepoints are calculated, and the entire waveform is iterated to calculate the steady-state waveform. The transient algorithm works by using all the equations at all the nodes in the circuit where the assumption is made that the sum of the currents is zero at every node and at every timepoint. Ohm's law is observed, and the equations look like `I=V*G` where `G` is the conductance. In general, the matrix form of the equations is constructed, so the system of simultaneous equations becomes `[G]*[V]=[I]` where `[G]` is a square matrix, and `[V]` and `[I]` are columns. To solve for [`V]`, the `[G]` matrix is inverted and this matrix multiplies the `[I]` matrix to solve for all the voltages on all the nodes. At the end of the pss analysis, all the solution matrices are saved as data to be passed to the small-signal analyses. The result is that the conductance is calculated and saved for every nonlinear component in the circuit at every timepoint.

For the following circuit:

© 1994-2020 All Rights Reserved.

Below is the I-versus-V curve with 3 different values for the input voltage. The center, positive peak of the input, and the negative peak of the input are shown. The input amplitude is small.

In the above figure, the amplitude of the 1 GHz signal causes small variations in the slope of the tangent of the nonlinear curve as the input signal is varied in amplitude. The slope is the conductance. Because the circuit has a nonlinear device, the conductance of the device varies slightly from timepoint to timepoint in the pss analysis. Because the input signal at 1.0GHz is small, the variation in conductance from timepoint to timepoint is small. The slope changes just a little bit as the input signal is moved over its range.

Now imagine that the 1GHz signal is 10dB larger. A 10dB change is a linear factor of about 3.16. The larger amplitude of the input causes a larger portion of the nonlinear device curves to be traversed, and thus the variation in conductance becomes larger. If a second input at

© 1994-2020 All Rights Reserved.

1.1GHz were applied to the circuit, this mixes with the fundamental and the harmonics of 1GHz and more strongly mixes up and down in frequency. This is illustrated below.

Therefore, the variation in the conductance of the devices from timepoint to timepoint when shooting is selected in pss, or the harmonic content if harmonic balance is selected in pss contains information about how strongly a circuit translates frequencies, and this is used by the pac small-signal analysis to calculate the small-signal conversion gain.

Overview of Simulation Capabilities

PAC has a choice called specialized analysis which is located near the bottom of the *Choosing Analyses* form. This sets the mode of the measurement to be either averaged over the whole pss waveform or at a specific timepoint in the pss waveform. Many different types of analyses can be performed, like conversion gain, distortion contribution, and instantaneous gain measurements. Below is a description of the capabilities for each specialized analysis setting.

Also note that in this section, just the concepts are discussed without any mention of how to do the setup in ADE Explorer. To see the ADE Explorer setup, see ADE Implementation on page 869.

© 1994-2020 All Rights Reserved.

Specialized Analysis=None

Down Conversion: Input Near the First harmonic

At the bottom of the pac *Choosing Analyses* form, there is a selection called *Specialized Analyses*. The default is *None*, and this is for measuring the conversion gain that is the average conversion gain with the large signal applied to the circuit (usually the LO or the sample clock) applied to the circuit. The mixing products that are produced are called sidebands. The sideband number is the harmonic number that is being mixed with to provide the output. A positive number means that the output mixing product is higher in frequency than the input frequency specified in the *Choosing Analyses* form. A negative number means that the output is lower in frequency than the input frequency. A diagram is shown below.

We tend to think of positive frequencies. The default is to reflect negative frequencies to positive frequencies.

Again, in pac, a single input frequency is applied at a time, and then the output mixing products are calculated through the highest harmonic of the periodic steady-state (pss) simulation specified by the *Maximum sideband* parameter. *Maximum sideband* defaults to seven for shooting and the total number of harmonics specified in the pss analysis for hb. Usually *Maximum sideband* is set to calculate the desired mixing products. This can be

© 1994-2020 All Rights Reserved.

accomplished by setting *Maximum sideband* or by using the select from range capability in the pac *Choosing Analyses* form.

Up-conversion: Input Near Zero Frequency

In a similar manner, up-conversion is shown below.

Frequency Sweep

This analysis is like AC, where the input frequency is swept. The input frequency range is specified in the pac *Choosing Analyses* form.

In addition to the frequency, there is a selection called *Sweep Type*. The choices are *relative* and *absolute*. When *absolute* is selected, the frequency range is used directly with no frequency conversion. When *relative* is selected, a *Relative Harmonic* field appears. The frequency sweep is shifted by the specified frequency of the harmonic in the pss analysis. For example, assume that the pss is at 1GHz, and a log sweep is desired above the first harmonic of the pss analysis. In this case, you need to select *relative* sweep, and specify 1 in the *Relative Harmonic* field. Next, type in 1K to 100M for the frequency range with a log sweep and 3 to 5 points per decade.

© 1994-2020 All Rights Reserved.

The sweep will start at 1G + 1K, and sweep to 1G + 100M using log spacing from 1K to 100M. This is illustrated below.

To use a log sweep below the first harmonic, use the same frequency range, and type in -1 as the relative harmonic. This is illustrated below.

© 1994-2020
All Rights Reserved.

Maximum Sideband

Sideband is the name of the PAC output mixing product. The sideband number is the pss harmonic number that is mixed with. If the sideband number is negative, the output frequency is mixed down in frequency from the input. If the sideband is positive, the output is mixed up in frequency.

Conceptually, an infinite number of mixing products are produced when a single input is applied to a nonlinear system. From the practical point of view, usually only a small number of mixing products need to be measured.

There is a property in the *Choosing Analyses* form called *Maximum sideband* that is used to define how many mixing products should be calculated. If *Maximum sideband* is zero, the pac analysis will only contain the frequency that appears without mixing or aliasing of any type. This is not equivalent to a linear AC analysis because pac takes into account the instantaneously varying nature of the LO or sample clock signal because of the large signal being present in the pss analysis.

When *Maximum sideband* is one, the outputs without frequency translation and the outputs that mix or alias with the first harmonic of the pss analysis are present in the pac analysis. When *Maximum sideband* is 10, the pac analysis includes mixing through the 10th harmonic of the pss analysis. When shooting is used for the pss analysis, the number of sidebands can be any number up to four times the number of harmonics specified in the pss analysis. When harmonic balance is selected, up to the number of harmonics in the pss analysis can be selected.

Setting Harmonics and Sidebands

Shooting

The pss analysis, when shooting is selected, has a minimum of 200 timepoints, so it inherently has frequency domain content through the 100th harmonic of the pss beat frequency. Maximum harmonic is a pss post-processing property that specifies how many harmonics to calculate in the Fourier transform of the waveform in pss. If the number of harmonics is zero, it just means that no frequency domain information should be calculated in the simulation. If the number of harmonics is 10 or less, there is no effect on the PSS waveform. When maximum harmonics gets larger than 10, the minimum number of timepoints in the pss analysis is increased. 20 timepoints at the frequency of the highest harmonic are forced. Thus, if maximum harmonics is raised, the pss waveform gets more accurate at the cost of longer runtimes and more memory required to run the pss simulation because more timepoints are forced in the pss simulation.

© 1994-2020 All Rights Reserved.

In order for the pac analysis to correctly measure conversion gain, the wave shape at the highest mixing harmonic needs to be accurate, so with the default settings in shooting pss, up to 40 sidebands can be accurately calculated in the pac analysis. In most cases, only a small number of sidebands needs to be calculated, and so with shooting, you can count on the pac result to be accurate.

If you need more than 40 sidebands when you are using shooting pss, you need to set the *maxacfreq* option in the pss analysis to the pss beat frequency times the number of sidebands requested in the pac analysis, or alternatively, set the number of pss harmonics to the number of sidebands divided by four. Either method works to set the maximum timestep in the pss analysis as required for the pac analysis to be accurate.

Harmonic Balance

When *Harmonic Balance* is selected in the pss analysis, only the harmonics specified in the pss *Choosing Analyses* form exist in the solution. In the pac analysis, because only the harmonics specified in the pss-hb *Choosing Analyses* form exist, the maximum sideband term can only be the number of harmonics specified in the pss analysis or less. Usually when *Harmonic Balance* is selected in the pss analysis, the *Maximum sideband* field in pac usually should be left blank which calculates all the output mixing products.

Note that with harmonic balance, the number of harmonics (and oversample factor if you have non-sinusoidal waveforms) in the pss analysis needs to be determined. Start with a number of harmonics in the pss based on an estimate of how many harmonics the circuit actually produces with your input waveform, and run the pss and pac analyses. Remember that for non-sinusoidal waveforms, oversample factor also needs to be set greater than one. Run the analysis and do the desired measurement. Now increase the number of harmonics by roughly 50% and run the simulation again. If the answer does not change, then you had enough harmonics originally, and you might actually be able to decrease the number of harmonics from the original setting. If you have non-sinusoidal waveforms, increase the oversample factor before increasing the number of harmonics to find the minimum value for both oversample factor and harmonics.

Modulated

Modulated pac is typically used for measuring AM to PM measurements in driven circuits and oscillators. In this case, the input is usually the power supply ripple at low frequency, and the desired output is near the first harmonic of the output. Modulated is an averaged measurement over the whole pss waveform (shooting) or all the harmonics (harmonic balance). Pac apples a modulated input signal to the circuit, and is able to measure a modulated output signal from the circuit. The types of modulation are Single Side Band (SSB), AM, and PM.

© 1994-2020 All Rights Reserved.

A diagram is shown below for SSB input.

Note that although the input occurs at one frequency, the AM output can be measured directly from the single input.

When AM is applied at the input, the following figure is shown.

© 1994-2020
All Rights Reserved.

The difference is that in addition to the input at low positive frequency, another input (to form an AM signal) is applied at a low negative input. This second input adds to the output signal near the first harmonic.

© 1994-2020
All Rights Reserved.

For PM, the negative input amplitude is negative, as shown below.

By running all three inputs, the SSB/AM/PM input to SSB/AM/PM output can be calculated. This is a setup choice in the *Choosing Analyses* form.

In the *Choosing Analyses* form, the input frequency range, and the output harmonic must also be specified.

All the plot functions are implemented in the *Direct Plot Form*.

Rapid IP2/IP3 (AC and PAC)

Rapid IP2 and Rapid IP3 have been added to the ac and pac analyses. First, a provision has been made to allow two input frequencies. Second, the ac and pac analyses have been enhanced so that harmonics of those signals and the intermodulation distortion can be calculated. Doing this in a small-signal analysis is much faster than doing this in a large-signal analysis because the large-signal effects are ignored.

The Rapid IP2 and Rapid IP3 that is available in the ac *Choosing Analyses* form allows the calculation of IP2 and IP3 for amplifiers. In pac, Rapid IP3 and Rapid IP2 are provided for circuits that translate frequencies, whether that translation is an up or down conversion. In this case, the signal that is causing the frequency translation (the LO or the sample clock) is applied in the pss analysis first in order to characterize the nonlinearity of the system so that

© 1994-2020 All Rights Reserved.

the correct IP2/IP3 measurement can be made. Note that pac and ac are small-signal analyses, so the limitation is that the measurement is a small-signal IP3 or IP2. If you need a large-signal IP2 or IP3 measurement, use a large-signal analysis like hb or qpss to make the measurement. Note that when you are using Rapid IP2 or Rapid IP3 in the ac analysis, either a SpectreRF license is required, or an additional mmsim token is required to run the analysis.

Make sure that the input level for the IP3 or IP2 measurement is in the small-signal range. Rapid IP2 and IP3 have exceptionally small numerical noise floors, therefore, it is recommended that you specify a level that is too small, rather than too large. Rapid IP2 and Rapid IP3 work by evaluating the device equations for nonlinearity. Do not specify one volt as the input amplitude because this is generally in a large-signal range, and usually causes the devices to reach points where the device currents are deliberately linearized, thus making the IP2/3 measurement more linear than it actually is.

The plot functions for Rapid IP2 and Rapid IP3 are implemented in the *Direct Plot Form*.

1. Select *Results - Direct Plot - Main Form*. In the Direct Plot Form.
2. Select *ac* or *pac* from the *Analysis* section.
3. Select *Rapid IP2* or *Rapid IP3* (depending on which was set up) from the *Function* section, and click *Plot*.

The X and Y points are shown for the intercept.

The X value is input-referred IP2/IP3, and the Y is output-referred IP2/IP3.

Note that if you have a switched-FET mixer and you are using the BSIM3 or BSIM4 models, due to a model limitation, the IP3 calculation from rapid IP2/IP3 will be incorrect. In this case, the PSP model is recommended. If a PSP model is not available, use hb or qpss with an input amplitude that is as large as possible while still being in the small-signal range for your circuit or switch to a 1dB gain compression measurement, The compression measurement is much more accurate than the IP3 measurement for switched FET mixers.

Compression and IP2 distortion summary

The distortion summaries measure distortion for an amplifier in ac or in circuits that have frequency translation in pac. The idea is to find which devices in a signal path contribute relatively more or less to the distortion. Since IP3 and compression are related mathematically, the compression distortion summary provides information about which devices contribute the most amount of third order intermodulation distortion. The IP2 distortion summary measures which devices contribute to the second order distortion. Note that when you are using the distortion summaries in the ac analysis for amplifiers, either a SpectreRF license is required, or an additional mmsim token is required to run the analysis.

© 1994-2020 All Rights Reserved.

Both distortion summaries use the *PAC Magnitude* property in the input port. Make sure that the *PAC Magnitude* (in volts or dBm) is actually in the small-signal range, (At least 10 dB smaller than the 1dB gain compression point) or the results will not be accurate. This is because the pac results do not take into account the large-signal limits like slew-rate or power supply voltage and the device models deviate from the ideal equations at very large amplitude by just continuing a linear slope above a certain amplitude. That amplitude depends on the device model parameters and is likely to be reached with a 1 volt input amplitude. The compression distortion summaries have an extremely low numerical noise floor, so it is better to set an input amplitude that is too small, than to set it too large.

The distortion summaries provide a list output, not a plot output. Accordingly, instead of selecting *Results - Direct Plot - Main Form*, select *Results - Print - PAC Distortion Summary*.

For the compression distortion, you get the actual gain with the distortion of each device by itself divided by the ideal small-signal gain provided by the pac analysis. In addition, the amplitude of the first three harmonics of the linear output frequency is calculated.

For the IM2 distortion summary, you get the total IM2 output amplitude that results from having only a single device in the circuit.

Sampled

Sampled is provided in order to calculate the instantaneous conversion gain at a voltage threshold crossing in the pss waveform or at specific timepoints in the pss simulation. Note that this analysis is fundamentally different than the analyses discussed earlier. All the previous analyses measured the average conversion gain that is produced when all the harmonics are taken into account (hb) or all the timepoints are taken into account (shooting). In sampled, an instantaneous measurement at a single point in time (or a list of times) of the LO or clock waveform is made.

Note that when harmonic balance is selected, sampled pac is the same as in shooting, but the time-domain waveform is the ifft of the harmonics calculated by the harmonic balance analysis. This is useful for measuring the conversion gain from the power supply to the oscillator output at a threshold crossing, or at specific times in the LO waveform to the output of a mixer.

When shooting is selected, a minimum of 200 timepoints exist in the solution waveform, and shooting automatically makes the timestep smaller in areas of rapid change, thus, there is likely a datapoint in the pss analysis that is very near the voltage level you specify as a threshold. In harmonic balance, the timepoints in the ifft depend on the number of harmonics and the oversample factor. Set a large enough number of harmonics and/or set a large

© 1994-2020 All Rights Reserved.

enough oversample factor to make sure that there is a timepoint in the ifft that is very near the actual threshold value you specify in the *Choosing Analyses* form.

© 1994-2020 All Rights Reserved.

ADE Implementation

Note: In general, the examples in this section show shooting pss. The examples work for harmonic balance pss unless otherwise noted. This is done mostly to show shooting. Harmonic balance has its own section in Chapter 3, “Frequency Domain Analyses: Harmonic Balance.”.

The settings in the PAC form can be slightly different for shooting and harmonic balance, and the method for both is covered in each section.

Circuit, Input port Setting, and ADE Explorer Setup for all the Examples in This Section

Consider the following circuit:

© 1994-2020
All Rights Reserved.

This above circuit is an LNA and a mixer for a 2.4GHz application that is implemented in CMOS. The input source (a port in Cadence terminology) has the values for its frequency and amplitude set to variable names. The PAC magnitude property is set to 1 volt in order to allow easy conversion gain measurements. The reason for the variables is that the frequency and amplitude can be changed in the ADE Explorer environment without changing the schematic.

© 1994-2020 All Rights Reserved.

The *Edit Object Properties* form for the input port is shown below.

© 1994-2020
All Rights Reserved.

ADE Explorer has the variables set to appropriate values, as shown below.

Note that the variables fb1 and fb2 (which set the input frequency of the input port) are set to zero, which effectively disables the RF input. Only the LO signal is applied to the circuit for the pss analysis.

Using PAC Analysis for Conversion Gain Measurement

Set up the pss analysis with just the large signal that causes the frequency translation to occur, which is generally the LO signal or the sample clock. For more information, refer to Overview of Periodic Steady-State (pss) Analysis on page 647. Regarding whether to use shooting or harmonic balance, pick the analysis that produces the smallest runtime. In

© 1994-2020 All Rights Reserved.

general, shooting is recommended for systems that have sharp edges, and harmonic balance is recommended for systems that are more sinusoidal.

© 1994-2020
All Rights Reserved.

Set up the pac *Choosing Analyses* form as follows:

© 1994-2020
All Rights Reserved.

1. Specify the input frequency range. In this case, the *Sweeptype* is set to *absolute*, which means that there is no frequency shift applied to the input frequency. If a log sweep above the first harmonic is desired, set the *Sweeptype* to *relative*, and specify one as the relative harmonic number. The input frequency range would be 1MHz to 100MHz to get the same frequency range as in the example above. Relative shifts the frequency specified the harmonic number times the pss frequency.
2. For pac, the *Maximum sideband* field should be set to the highest harmonic of the pss analysis that mixing products are desired to be calculated. This is generally determined by the application. If you only need one or two output mixing product, choose *Select from range*, and highlight the sidebands you want to calculate. Setting *Maximum sideband* for pac is usually not critical as long as enough sidebands are calculated to measure the desired output mixing product (called a sideband). The *Direct Plot Form* allows the plotting of individual output mixing products as will be shown later. As a general recommendation, set the number of sidebands equal to the number of pss harmonics.
3. For a conversion gain measurement, set the *PAC Magnitude* to 1 volt on the *Edit Object Properties* form, as shown earlier. This is always volts peak.
4. Now run the simulation. In ADE Explorer, select *Simulation - Netlist and Run*.

© 1994-2020 All Rights Reserved.

5. After the simulation completes, select *Results - Direct Plot - Main Form*. The *Direct Plot Form* is displayed.

Mixing products are calculated for mixing the pac input frequency range with the harmonics calculated in the pss analysis. Harmonics of the pac input and harmonics of the output mixing products are not calculated because pac is a small-signal analysis. Because it is small-signal, the harmonics of the input and the harmonics of the mixing products are so small that they do not matter.

© 1994-2020 All Rights Reserved.

To measure the outputs you desire, place a marker at the desired output frequency of interest and read the conversion gain, as shown below.

To plot a single output sideband (mixing product), in the *Direct Plot Form*:

© 1994-2020
All Rights Reserved.

1. Select *sideband* instead of *spectrum*, and select the desired output mixing product form the list as shown below.

© 1994-2020
All Rights Reserved.

2. Select the output net in the schematic, or click the *Replot* button in the *Direct Plot Form*. The waveform tool displays the output, as shown below.

3. Position a marker by moving the cursor to the desired output frequency and type `m`. The marker appears, as shown in the figure above.

4. To position the marker at an exact frequency, place it as above, and then move your cursor over the intercept point on the trace. Click and hold the right mouse button, move to *Marker Properties*, and release the mouse button. The *Point Marker Properties* window appears, as shown below.

© 1994-2020
All Rights Reserved.

5. Type the frequency in the *Position* field, and click *OK* or *Apply*.

© 1994-2020
All Rights Reserved.

Sampled Circuits and Switched-capacitor Filters

For sampled circuits or for switched-capacitor circuits, always use shooting pss because the clocking waveforms have rapid transitions. Shooting captures these transitions in a natural and accurate way.

In pac, generally only the sideband with zero frequency translation is desired, so *Maximum sideband* can be set to zero. A warning is produced in the Spectre output window stating that the number of sidebands is lower than anticipated. This warning can be ignored.

Note that the average gain is determined by pac. In the sampling circuit, this is generally very near unity gain, but for the switched-capacitor circuit without a sample and hold, the gain is generally a bit less than half amplitude which is about -6dB. This is because the output waveform follows the input waveform for a little less than half the time, and the capacitors are reset by the switches the other half the time.

If you desire to see the aliasing at different harmonics of the clock, use pxf analysis. For example, imagine a sampling frequency of 1MHz. In pac, with the zero sideband only, the input and output frequencies are always the same. Therefore, if the input is at 1.1MHz in pac, the output is also at 1.1MHz. Generally, the term that is interesting is the term that aliases down from 1.1MHz to 100KHz which is not measured by the pac analysis (The input and output frequencies are the same in the zero sideband.) To see the aliasing at the different harmonics, use pxf with an output frequency of near zero to 500KHz, and specify *Maximum sideband* to the harmonic number desired. If the *Maximum sideband* is greater than 40, either set the pss option *maxacfreq* to the number of sidebands times the pss beat frequency, or set the number of harmonics to the number of sidebands divided by four.

© 1994-2020 All Rights Reserved.

Commonly Used PAC Options

Freqaxis

© 1994-2020 All Rights Reserved.

freqaxis specifies whether you want to see the negative frequency axis or not.

The analysis calculates the output frequencies and amplitudes, so *out* and *absout* are reasonable choices. *out* displays the negative frequency axis. *absout* (absolute value of the output) displays positive output frequencies. This is the default.

If you select *in*, all the outputs at the different frequencies are plotted on the same input frequency range scale. This is not recommended. The default is *absout*.

Below is an example of *freqaxis* set to *out*. Note that the data is the same as before. It is just displayed with the negative frequency axis present. The negative frequencies have not been reflected up to the positive frequency domain.

Below is an example with the *freqaxis* option set to *in*. All the outputs at different frequencies are plotted on the input frequency scale. The frequency can be calculated by selecting the appropriate sideband from the list on the left side of the waveform tool, and then adding the sideband number times the pss beat frequency to the frequency displayed on the X axis.

© 1994-2020 All Rights Reserved.

Tolerance

Leave this option at the default value.

Pac uses an iterative solver to calculate the output amplitudes. Any iterative solver needs an error tolerance to specify when to stop iterating because the solution is accurate enough, and the tolerance option specifies that accuracy for pac. For shooting, the default tolerance is 1e-9. For driven circuits where HB is the pss engine, the default is 1e-6. For oscillators where hb is selected for the pss engine, the default is 1e-4.

Solver

The solver option is used only for driven circuits. The default solver is the turbo solver.

When hb is used as the engine in pss, leave this option at the default.

© 1994-2020 All Rights Reserved.

When shooting is used, sometimes when the pac input frequency is very close to the frequency of one of the harmonics in the pss, warning messages will appear in the pac output warning that the accuracy might not be good enough. If you see these messages, select the std solver, which has better ability to handle frequencies that are very close to a harmonic in the pss, but which takes longer to run than the turbo solver.

Hbprecond_solver

This option is only available only when harmonic balance is selected in PSS.

The basic solver is the only solver available in standard Spectre. Autoset is the default solver when APS is used. This solver is faster, but occasionally stagnates. When stagnation is detected, APS automatically switches to the basic solver, and prints a message in the Spectre output window.

Oscsolver

The oscsolver option is used only for oscillators. The default solver is the *turbo* solver.

When hb is used as the engine in pss, leave this option at the default.

When shooting is used, sometimes when the pac input frequency is very close to the frequency of one of the harmonics in the pss, warning messages appear in the pac output warning that the accuracy might not be good enough. If you see these messages, select the *std* solver, which has better ability to handle frequencies that are very close to a harmonic in the pss, but takes longer to run than the *turbo* solver.

For very large oscillator designs, the ira oscsolver may take less memory and run faster with no loss in accuracy, however, it is less robust for convergence compared to *turbo* or *std*.

Gear_order

Do not change this option.

Resgmrescycle

Leave this option at the default.

© 1994-2020 All Rights Reserved.

Outputperiod

Do not use this option.

AdditionalParams

In some cases when shooting is used in pss on an oscillator circuit measurements near a pss harmonic may be questionable. If this occurs, type osc_version=dts in the addionalParams field to solve the issue.

additionalParams is typically used for new features that are being beta tested.

For more information about the other options, type `spectre -h pac` at the command prompt in a Unix shell window

PAC With Multiple PSS Inputs

This is generally used when the conversion gain is desired with a large blocker signal present at the RF input. The basic strategy is to apply the signals that cause the nonlinearity to occur (the LO and the blocking signal) in the pss, and then use pac to measure the conversion gain when the blocker is present. In many cases, the blocker power is swept in order to measure the degradation in conversion gain when the blocker gets large.

Multiple input frequencies might be able to be run in the PSS, depending on the ratio between the highest input frequency divided by the pss beat frequency. If this ratio is about 25 or less, then pss can be used for the simulation. If the ratio is greater than 25, consider using harmonic balance (hb) or qpss to run the simulation because it is very likely to take less time to run.

To set up the second input, the variable fb1 was set to 2.5G in the ADE Explorer window, as shown

© 1994-2020 All Rights Reserved.

below. This variable sets the frequency of the first source in the port component that supplies the RF signal. In this case, the LO is at 2.4GHz and the RF is at 2.5GHz.

© 1994-2020 All Rights Reserved.

The pss was set up with 175 total harmonics of 100MHz to allow harmonics through the seventh harmonic of the LO signal at 2.5GHz to be calculated. For more information on pss, Overview of Periodic Steady-State (pss) Analysis on page 647.

© 1994-2020 All Rights Reserved.

© 1994-2020
All Rights Reserved.

Remember that the frequency range is the input frequency range. In this example, the frequencies just above the first harmonic of the LO are swept.

The pac form is set up to calculate the mixing products for all the harmonics that are calculated in the pss analysis. When shooting is used in the pss analysis, up to four times as many sidebands can be specified in pac as harmonics are specified in pss. When harmonic balance is used in pss, *Maximum sideband* can be set up to the number of harmonics in the pss analysis. Sidebands are the mixing products that are produced when the input signal mixes with the harmonics in the pss analysis.

© 1994-2020
All Rights Reserved.

Now run the analysis. The spectral plot below from the pss analysis shows that because there were more input frequencies in the circuit, many more harmonics are produced by the circuit, which is as expected.

As shown below, because there are many more pss harmonics to mix with, many more mixing products (sidebands) are produced by pac.

© 1994-2020
All Rights Reserved.

© 1994-2020
All Rights Reserved.

The next figure shows a zoomed-in area near zero frequency. Only the mixing products larger than -100 dB are shown.

Note that the index in the pac analysis is the frequency shift from the input frequency to the output frequency in multiples of the pss beat frequency. The output highlighted above mixes with the 24th harmonic of 100MHz, which is the LO signal in the circuit. This is the main output mixing product when RF mixes with LO.

Note that the RF input in the pac analysis at 2.4001G to 2.45G can also mix with 2.5G (The additional RF signal.) When the RF input is at 2.4001G the IF output is 99.9MHz when it mixes with the other RF tone at 2.5GHz. As the input frequency in the pac sweeps up, the output frequency sweeps down to 50MHz. Since it is mixing with 2.5GHz, the output frequency is downshifted by -25*100MHz, thus the sideband number shown in the legend at the left is -25. This is shown below.

© 1994-2020
All Rights Reserved.

Rapid IP3/IP2

Rapid IP3 and Rapid IP2 are available in the ac and the pac *Choosing Analyses* forms. Rapid IP3 in the ac form is for measuring IP3 of amplifiers where there is no frequency translation, and Rapid IP3 in the pac form allows the measurement of IP3 where frequency translations are present. Rapid IP3 and IP2 measure small-signal IP3 and IP2.

The example below is for an LNA-mixer circuit.

© 1994-2020
All Rights Reserved.

1. First set up the pss form with just the LO applied, as shown below. For more information, see Overview of Periodic Steady-State (pss) Analysis on page 647. Both the shooting and harmonic balance pss engines are supported in pac.

© 1994-2020 All Rights Reserved.

© 1994-2020
All Rights Reserved.

In the example below, the LO is at 2.4GHz, and the RF inputs are at 2.401GHz and 2.4012GHz. This makes the main mixing products at the output of the mixer at 1MHz and 1.2MHz. The third order intermods are at 800KHz and 1.4MHz. Because the frequencies are similar, either the 1 or 1.2 MHz products can be set as the linear frequency, and either the 800KHz or 1.4MHz term can be used for the third order frequency. Generally, the smallest amplitude linear term and the largest amplitude third order term are used for the IP3 measurement. If IP2 was desired, the second order product is at the difference frequency between the 2 RF inputs (200KHz in this example). The first order terms are the same as IP3.

2. Select the *pac* analysis. At the bottom of the form in the *Specialized Analysis* section, select *Rapid IP3* or *Rapid IP2*, depending on what you want to measure. The entries for the forms are identical, except for the frequency of the intermodulation product. For switched FET mixers, make sure that the psp model is used. Incorrect results are produced for BSIM3 and BSIM4 models due to a limitation in the model. If you need to measure IP3 with a BSIM3 or 4 model, use hb analysis with as large an amplitude as possible while maintaining the amplitude in the small-signal region.

3. Define the input port, input frequencies, input power, the frequencies of the linear and second or third order products, and the output node. An example for Rapid IP3 is shown below. Make sure that the input power is in the small-signal range. (At least 10dB smaller than the compression point)

4. Select the output net in the circuit.

   Note that for amplifiers, the pss analysis does not need to be set up. Just use the ac *Choosing Analyses* form which has the same appearance as the pac *Choosing Analyses* form shown below. The linear output frequency can be either of the two input

© 1994-2020 All Rights Reserved.

frequencies. The third-order frequency can be the frequency spacing above or below the RF input frequencies.

© 1994-2020
All Rights Reserved.

5. Leave the *Maximum Non-linear Harmonics* field blank. The default value is appropriate for the Rapid IP2/IP3 calculation.

6. Run the simulation. When the simulation is complete, select *Results - Direct Plot - Main Form*.

7. Select *pac* from the *Analysis* section.

8. Select *Rapid IP3* from the *Function* section.

   Rapid IP2 is shown if an IP2 analysis is run.

9. Click *Plot*.

   The waveform is displayed. The marker shown at the intercept point shows the input and output-referred IP3 or IP2.

© 1994-2020 All Rights Reserved.

Compression Distortion Summary

The compression distortion summary is provided in order to get a relative measure of which components in a gain path contribute more or less compression. Because compression and IP3 are related mathematically, the compression distortion provides information about which components contribute to IP3.

Note that the Compression Distortion Summary is available in the ac and the pac *Choosing Analyses* forms. Compression Distortion Summary in the ac form is for measuring which components add compression in amplifiers where there is no frequency translation, and Compression Distortion Summary in the pac form allows the measuring of components that contribute to compression where frequency translations are present in the circuit.

Note that for sampling circuits or for switched-capacitor filters, the sample clock(s) are applied in the pss, and the small-signal input Is applied in the pac analysis.

© 1994-2020
All Rights Reserved.

The *Contributor Instances* field is provided so that you can select which components in the circuit should be considered in the analysis. If the *Contributor Instances* field is left blank, all the nonlinearities in your circuit are run.

The following example is for an LNA-Mixer circuit.

© 1994-2020 All Rights Reserved.

1. First set up the pss analysis with just the LO or sample clock applied, as shown below. For more information, see Overview of Periodic Steady-State (pss) Analysis on page 647.

© 1994-2020 All Rights Reserved.

2. Now select pac analysis.
3. Select *Compression Distortion Summary* from the *Specialized Analyses* drop-down list.

   The following notice appears to remind you to set the PAC amplitude parameter in the source in the schematic to a value somewhere in the small-signal range. This is generally at least 10dB smaller than the 1dB gain compression point.

4. If you leave the *Contributor Instances* field blank, all the nonlinear devices in your circuit will be run. If you want a subset, click *Select*, and select the instances you want on the schematic.
5. Specify the input frequency in the *Frequency of Input Source* field.
6. Specify the output frequency in the *Frequency of Linear Output Signal* field.
7. Leave the *Maximum Non-Linear Harmonics* field blank. The default value is *optimum* for the compression distortion measurement.
8. Select the output node in the circuit. When the *Out-* field is left blank, ADE Explorer automatically assigns this to the global ground node.

© 1994-2020 All Rights Reserved.

The completed form appears similar to the one shown below.

© 1994-2020
All Rights Reserved.

9. In the *Edit Object Properties* form, make sure that the *PAC magnitude (dBm)* is set in the small-signal range. In this example, it is set to -30Bm.

10. If you make changes to your schematic that you want for the distortion summary, but you do not want to save them to the circuit file, make the desired changes and then select *Check - Current Cellview* in the schematic window. Later, if you want to save them, just

© 1994-2020 All Rights Reserved.

select *Check* and *Save*. If you do not want them, quit from the schematic window, and do not save the changes.

ADE Explorer will netlist from the checked view, so it will get the changes.

11. When you are done, run the simulation.

12. When the simulation completes, select *Results - Print - PAC Distortion Summary*.

© 1994-2020 All Rights Reserved.

The summary is displayed below.

Results Display Window

Window Expressions Info Help cadence

```
PAC Compression Distortion Summary

Results in        pac_distortion

                                        Nonlinear Mag(Phase)[V(Deg)] at 1st 2nd & 3rd harm of linear freq
Instance          Distortion(dB)    freq=1e+06        freq=2e+06        freq=3e+06
Total             -379.6m           3.772m(120.7)     3.205u(-25.34)    820.6u(143.2)
I33.I1.I0.PM3     323.8u            49.17u(26.52)     168.8u(-18.23)    13.62u(48.44)
I33.I1.I0.PM5     323.6u            49.17u(26.52)     168.8u(161.7)     13.61u(48.38)
I33.I1.I1.I2.PM12    407.2n            4.432n(-80.79)    1.896u(75.2)      425.7n(-147.7)
I33.I1.I1.I2.PM11    54.07n            1.543n(9.495)     1.993u(-66.2)     416.9n(-105.4)
I33.I1.I1.I2.NM3    17.85n            262.1p(-105.9)    82.15n(130)       86.28n(-115.6)
I33.I1.I1.I2.NM33    6.177n            364.9p(-139.8)    89.75n(-63.47)    85.38n(-120.9)
I33.I1.I0.I2.NM1    1.991n            31.11p(-109.1)    7.611n(-40.62)    3.808n(172.1)
I33.I1.I1.I2.NM1    1.885n            20.29p(-40.29)    6.261n(-140.7)    4.259n(-96.86)
I33.I1.I0.I2.NM2    1.872n            29p(-108.7)       8.965n(130.8)     3.756n(172.2)
I33.I1.I1.I2.NM2    1.837n            19.23p(-45.61)    6.105n(45.58)     4.28n(-96.91)
I33.I1.I1.I2.NM32    44.34p            1.887p(-135.9)    287.5p(71.11)     93.82p(141.3)
I33.I1.I1.I2.NM0    9.433p            1.284p(-145.4)    82.4p(-73.47)     93.26p(142.3)
I33.I1.I0.I0.I2.R0.R9    1.929f            0(0)              0(0)              0(0)
I33.I1.I0.I0.I2.R0.R8    1.929f            0(0)              0(0)              0(0)
I33.I1.I0.I0.I2.R0.R7    1.929f            0(0)              0(0)              0(0)
I33.I1.I0.I0.I2.R0.R6    1.929f            0(0)              0(0)              0(0)
I33.I1.I0.I0.I2.R0.R5    1.929f            0(0)              0(0)              0(0)
I33.I1.I0.I0.I2.R0.R4    1.929f            0(0)              0(0)              0(0)
I33.I1.I0.I0.I2.R0.R3    1.929f            0(0)              0(0)              0(0)
I33.I1.I0.I0.I2.R0.R2    1.929f            0(0)              0(0)              0(0)
I33.I1.I0.I0.I2.R0.R1    1.929f            0(0)              0(0)              0(0)
I33.I1.I0.I0.I2.R0.R0    1.929f            0(0)              0(0)              0(0)
I33.I1.I0.I0.R14.R19    1.929f            0(0)              0(0)              0(0)
I33.I1.I0.I0.R14.R18    1.929f            0(0)              0(0)              0(0)
I33.I1.I0.I0.R14.R17    1.929f            0(0)              0(0)              0(0)
I33.I1.I0.I0.R14.R16    1.929f            0(0)              0(0)              0(0)
I33.I1.I0.I0.R14.R15    1.929f            0(0)              0(0)              0(0)
I33.I1.I0.I0.R14.R14    1.929f            0(0)              0(0)              0(0)
I33.I1.I0.I0.R14.R13    1.929f            0(0)              0(0)              0(0)
I33.I1.I0.I0.R14.R12    1.929f            0(0)              0(0)              0(0)
I33.I1.I0.I0.R14.R11    1.929f            0(0)              0(0)              0(0)
I33.I1.I0.I0.R14.R10    1.929f            0(0)              0(0)              0(0)
I33.I1.I0.I0.R14.R9    1.929f            0(0)              0(0)              0(0)
I33.I1.I0.I0.R14.R8    1.929f            0(0)              0(0)              0(0)
```

41 Show Output

- The first column is the instance name in the schematic.
- The second column is the gain with the nonlinearity of that single device divided by the ideal gain from the ac or pac analysis.

  Gain compression is shown as a negative number. Gain expansion is shown as a positive number.

© 1994-2020 All Rights Reserved.

- The third, fourth, and fifth columns are the amplitudes of the first, second, and third harmonic of the linear output frequency with just the nonlinearity of that single device taken into account.

Sampled Circuits and Switched-capacitor Filters

For sampled circuits or for switched-capacitor circuits, always use shooting pss because the clocking waveforms have rapid transitions. Shooting captures these transitions in a natural and accurate way.

In pac, set up the compression distortion form just as you would for the mixer example above. Note that the linear output frequency is the input frequency. Otherwise, the setup is exactly the same as the setup shown above.

IM2 Distortion Summary

The steps for the IM2 distortion summary are very similar to the Compression Distortion Summary in the previous section. The differences are noted below.

- The IM2 distortion summary provides the output voltage (magnitude, real, and imaginary parts) of the second order distortion signal for each component in the schematic by itself.
- The IM2 distortion summary does not provide a selection mechanism for the devices in the circuit. All the devices are run in the IM2 distortion summary.
- The output format is slightly different. The first column is the instance name in the schematic. The second column is the magnitude of the IM2 product that is produced by

© 1994-2020 All Rights Reserved.

that single component in volts. The third column has the real and imaginary parts of the voltage. A sample is shown below.

```
PAC IM2 Distortion Summary

Results in        pac_im2distortion

Instance          Distortion(V)[Magnitude Complex]
Total             56.8138n    complex(56.8132n, 279.356p)
/I33/I1/I0/PM4    1.45004m    complex(1.45003m, -2.96617u)
/I33/I1/I0/PM3    1.45003m    complex(1.45003m, -2.96618u)
/I33/I1/I0/PM2    1.45002m    complex(-1.45001m, 2.96629u)
/I33/I1/I0/PM5    1.45001m    complex(-1.45001m, 2.96628u)
/I33/I1/I1/PM4    1.43768m    complex(1.43768m, -2.95021u)
/I33/I1/I1/PM3    1.43768m    complex(1.43767m, -2.95019u)
/I33/I1/I1/PM2    1.43767m    complex(-1.43767m, 2.95025u)
/I33/I1/I1/PM5    1.43767m    complex(-1.43767m, 2.95024u)
/I33/I1/I1/I2/NM3    144.198n    complex(-97.8239n, -105.941n)
/I33/I1/I1/I2/NM33    144.197n    complex(97.8234n, 105.94n)
/I33/I1/I0/I2/NM3    140.951n    complex(-96.7777n, -102.476n)
/I33/I1/I0/I2/NM33    140.951n    complex(96.7775n, 102.476n)
/I33/I1/I1/I2/PM11    10.0589n    complex(-7.44012n, -6.7695n)
/I33/I1/I1/I2/PM12    10.0589n    complex(7.44012n, 6.76949n)
/I33/I1/I0/I2/PM11    10.0337n    complex(-7.6139n, -6.53482n)
/I33/I1/I0/I2/PM12    10.0337n    complex(7.6139n, 6.53481n)
/I33/I1/I1/PM1    5.07326n    complex(-5.07326n, 4.89652p)
/I33/I1/I0/PM0    4.36043n    complex(4.36039n, 18.022p)
/I33/I1/I1/PM0    3.67355n    complex(3.67349n, -21.7191p)
/I33/I1/I1/NM0    3.00179n    complex(-3.00172n, 20.0364p)
/I33/I1/I1/NM1    2.61043n    complex(2.61009n, -41.8165p)
/I33/I1/I0/NM0    1.95443n    complex(-1.95432n, -20.0203p)
/I33/I1/I0/NM1    1.64469n    complex(1.64469n, 387.776f)
/I33/I1/I0/PM1    753.377p    complex(753.304p, 10.4897p)
/I33/I0/NM9       266.846p    complex(-266.8p, -4.96938p)
/I33/I1/I1/I2/NM2    226.574p    complex(-212.697p, -78.0755p)
/I33/I1/I1/I2/NM1    226.573p    complex(212.694p, 78.079p)
/I33/I0/NM12      165.221p    complex(165.216p, 1.39871p)
/I33/I0/NM10      126.074p    complex(125.993p, -4.53295p)
```


© 1994-2020 All Rights Reserved.

Sampled Circuits and Switched-capacitor Filters

For sampled circuits or for switched-capacitor circuits, always use shooting pss because the clocking waveforms have rapid transitions. Shooting captures these transitions in a natural and accurate way.

In pac, set up the IP2 distortion form just as you would for the mixer example above. Note that the linear output frequency is one of the input frequencies and the intermodulation frequency is the frequency spacing. Otherwise, the setup is exactly the same as the setup shown above.

Modulated PAC Analysis

Modulated pac analysis is usually applied to oscillators to measure AM-to-PM conversion from low-frequencies on the power supply (ripple) to the output near the first harmonic. It can also be applied to digital circuits for the same purpose. Although the shape of the transfer function is different, the steps are the same for both applications.

In the circuit, make sure that any PAC magnitude terms are removed on the input source (if there is one), and set the PAC Magnitude to 1 volt in the source that is used as the power supply. If you do not want to save the change permanently, just select *Check - Current View* in the schematic.

© 1994-2020 All Rights Reserved.

This example is for an oscillator. The circuit is shown below:

© 1994-2020
All Rights Reserved.

In ADE Explorer, set values for the power supply voltage and for the tuning voltage.

© 1994-2020
All Rights Reserved.

Set up the pss analysis. For more information on pss, see Overview of Periodic Steady-State (pss) Analysis on page 647.

© 1994-2020
All Rights Reserved.

In the pac *Choosing Analyses* form, first select *Modulated* in the *Specialized Analyses* section, as shown below.

© 1994-2020
All Rights Reserved.

Note that selecting *Modulated* causes the *Sweep type* to be set to *relative*. Since the gain from power supply ripple to the output is desired, the relative harmonic number is zero, and then the desired frequency range is entered. If the shooting engine is used, maximum sideband can be up to 4 times the number of harmonics in the pss analysis harmonics as shown above. Setting the *Sweeptype* to *relative* allows a frequency shift in multiples of the pss beat frequency to be applied to the input frequency. In this example, when the relative harmonic is zero, the input frequency range is not shifted at all in frequency.

Maximum sideband needs to be set larger than is needed for measuring the output frequency. In this case, the input signal is the ripple at low frequency, and the output is near the first harmonic. Because of this, *Maximum sideband* needs to be one or larger. In this case, the setting of *Maximum sideband* is set to the same number as the maximum harmonics setting in pss.

Input Type is set to *SSB/AM/PM* so that all the combinations of SSB, AM, and PM to SSB, AM, or PM can be calculated.

Next, the output frequency ranges to be calculated can be set. Note that a list of harmonics can be entered. The easiest way is to click the *Choose* button to the right of the field, and select the desired frequencies from a list in the *Choose Harmonic* window, as shown below.

© 1994-2020 All Rights Reserved.

The same process is used to set the input frequency, but only one frequency range is allowed.

Once the analysis has been set up, run the simulation. When the simulation completes, select *Results - Direct Plot - Main Form*.

© 1994-2020 All Rights Reserved.

The input defaults to *AM*. Select *PM* for the output. Since many output frequencies were calculated, you need to select the one you want to plot. In this example, the output just above the first harmonic is selected. Once this is done, select a net in the schematic.

© 1994-2020
All Rights Reserved.

The output is displayed. To make a measurement at a specific frequency, position a marker. The measurement is in radians per volt of input on the supply that has PAC magnitude set.

Sampled PAC Analysis

Pac sampled is for measuring the conversion gain at a specific instantaneous time in the pss waveform. The first step is to determine an appropriate threshold for the measurement by setting up the pss analysis, as shown below. Shooting is shown below, but this is also

© 1994-2020 All Rights Reserved.

available in harmonic balance. For harmonic balance, the waveform is an ifft of the harmonics.

© 1994-2020
All Rights Reserved.

Now run the analysis. When the analysis completes, select *Results - Direct Plot - Main Form* in ADE Explorer and plot the time-domain waveform on the output net, as shown below.

The waveform is shown below.

Note that 2.5 volts is a value near the middle of the waveform. This is chosen as the threshold for the calculation. Pick a threshold that is appropriate for your system.

Now set up the pac *Choosing Analyses* form.

1. At the top of the form, set up the input frequency range like normal.
2. At the bottom of the form, select *Sampled*. Usually the gain to a node is desired, so select voltage.
3. Click *Select* to the right of the *Net+* field and select the node in the schematic. Note that if the *Net-* field is left blank, the global ground node is used.
4. Specify the desired threshold voltage based on the time-domain waveform from the pss analysis
5. Choose an option from the *Crossing Direction* drop-down box. In the example, *all* is selected.

© 1994-2020 All Rights Reserved.

6. If you need more than 16 crossing points, type the number you need in the *Maximum Samples* field.

© 1994-2020 All Rights Reserved.

7. If you want to sample at specific times from the pss waveform, you can type in the times in the *Additional Timepoints* field.

© 1994-2020
All Rights Reserved.

8. Now run the simulation. When the simulation completes, open the *Direct Plot Form* and set it up, as shown below.

9. Select *pac* from the *Analysis* section.
10. Select *Voltage* from the *Function* section.
11. Select *spectrum* from the *Sweep* section.
12. Select *dB20* from the *Modifier* section.
13. Select the time you want from the list.
14. Select the output node in the schematic.

© 1994-2020 All Rights Reserved.

The waveform is plotted as shown below

To zoom in to the frequency of interest, double-click one of the numbers on the X Axis, and type in the output frequency range of interest. This is shown below.

To measure the conversion gain to the output, move the tracking cursor to the output frequency and read the gain or place a point marker by typing `m`.

© 1994-2020
All Rights Reserved.

© 1994-2020
All Rights Reserved.

Periodic Noise Analysis (*Pnoise*)

Small-Signal Versus Large-Signal Analysis for Noise

In most cases, noise is a small-signal problem in the real world. Consider an amplifier. One approach is to calculate the small-signal gain from the noise sources to the output, then calculate the frequency response of the noise source, and then add up the noise power at the output. This is done in the linear noise analysis. This runs quickly because the gain calculation just takes the tangent at the DC operating point for everything in the circuit. This ignores large-signal effects like slew-rate limits and clipping. Because noise in most circuits is actually very small compared to the output levels, this approach works well. Because it is a linear approach, frequency translations cannot be calculated.

When the input to the amplifier gets large enough that it starts to create nonlinearity, the frequency translations in the circuit need to be taken into account. Noise near the harmonics can mix up or down to the output frequency. In this case, the large signal needs to be applied in a large-signal analysis like periodic steady-state (pss) where the nonlinearity and harmonic levels can be calculated. After the pss runs, pnoise runs and is similar to linear noise except the frequency translations of the circuit are taken into account. Instead of calculating a single gain from the noise source to the output, both the amplitude of the noise and the gain for the noise sources at the significant harmonics needs to be calculated, and each noise source now has several contributions at the output from several different noise frequencies that mix or alias to the output frequency. The basic strategy is to apply the signals that cause nonlinearity in the pss analysis, and follow that with a pnoise calculation.

This same approach can be taken for mixers where the LO signal deliberately introduces nonlinearity for the purpose of frequency translation. The LO signal is applied to the mixer in the pss analysis, and pnoise follows that. This works with the mixer by itself, or in combination with other circuits, like an LNA and a baseband amplifier.

Many designs need to tolerate a large amplitude blocking signal. The basic strategy is to apply the LO and the blocking signal (and not the RF signal) in the pss analysis and follow with pnoise. Note that depending on the frequencies of the LO and the blocker, pss may not be appropriate. If the highest frequency divided by the periodicity of the system (the pss beat frequency) is about 25 or less, pss-pnoise is a reasonable approach. If the highest input frequency is 2.5GHz and the periodicity is 1MHz, use hb-hbnoise or qpss-qpnoise instead because they will require much less time and memory to run. This is because pss calculates all the harmonics of the pss beat frequency, and 2500 harmonics would be needed just to calculate the fundamental frequency of the 2.5GHz input. Both hb and qpss calculate just the mixing frequencies that are actually produced by the circuit, which is a much smaller number of harmonics. The blocker power can be swept in the pss analysis and pnoise is run after each

© 1994-2020 All Rights Reserved.

sweep value for the blocker power to measure the increase in noise with higher blocker power (desensitization).

For large-signal noise problems, or for systems that are not periodic, transient noise is provided. For more information on this, refer to the *Spectre® Circuit Simulator and Accelerated Parallel Simulator User Guide*. This approach requires specifying a maximum noise frequency to be analyzed, which in RF systems is usually quite high. Two timepoints are created in the period of noisefmax. The transient needs to be run over many noise points in order to characterize the noise present in the circuit and needs post-processing to see the results in the frequency domain. This usually requires much more time than using small-signal noise analyses.

Overview of Simulation Capabilities

The fundamental quantity calculated in pnoise is the output noise of the circuit with a large signal applied and including the noise translations caused by the nonlinearity of the circuit. In order to run pnoise, a periodic steady-state (pss) analysis needs to be run first in order to calculate the nonlinearity of the circuit and the amplitude of the harmonics that are present to mix with. Pnoise takes this information from the pss analysis and then for each noise source, it calculates the transfer functions from all the frequencies specified by the maximum sideband field in the *Choosing Analyses* form. Then it calculates the frequency response of the noise source itself. Next, it calculates the total noise power at the output for that source at all the noise frequencies that contribute at the output. Finally, it adds up all the noise power from all the sources at all the frequencies. Input-referred calculations require the selection of the input frequency range by setting the reference sideband. This selection is necessary because there are many transfer functions from the input to the output caused by mixing or aliasing from the different harmonics of the system. The passband frequency is also needed for the noise figure calculations.

Note that pnoise is a small-signal calculation that works for periodic systems. If the noise is large-signal, or the system is not periodic, transient noise should be used.

© 1994-2020 All Rights Reserved.

Example

Consider the double-balanced diode mixer below where a single-sideband noise figure measurement is desired.

The input source has variables set to define the frequency and amplitude of the two large-signal outputs. This is done so that the frequency and the amplitude can be set in the ADE Explorer window without changing the schematic.

© 1994-2020
All Rights Reserved.

The property list is shown below.

© 1994-2020
All Rights Reserved.

The variables have values set under the *Design Variables* section in the *Setup* pane of ADE Explorer. The easiest way to get the variables list is to generate a netlist from ADE Explorer or use *Variables - Copy From CellView*. This causes all the variable names to be entered in the variables section. Just click the value field for each variable and enter a value. Press <Enter> when you are done. In the figure below, the LO frequency set by the variable flo is 1GHz. This circuit just has the signal that causes the frequency translations to occur (the LO signal) applied. Note that 1.4 volts peak is +13 dBm. This is a common drive level for diode mixers.

The pss analysis needs to be run first in order to measure the amplitude of the harmonics and to characterize the nonlinearity of the system. From this, the noise folding from all the noise sources can be calculated in the pnoise analysis. For more information on the setup of the pss form, please refer to Overview of Periodic Steady-State (pss) Analysis on page 647.

Pss can be set to use either the shooting engine or the harmonic balance engine. The example below uses the shooting engine. Because shooting is a variation of the transient which automatically reduces the timestep in areas or rapid transition (where the diodes switch) and because the pss includes at least 200 timepoints in the solution, the PSS Shooting solution itself contains information to the 100th harmonic of 1GHz. For the pnoise to be accurate, at least 5 timepoints are needed at the highest frequency that is being

© 1994-2020 All Rights Reserved.

calculated in the pnoise. Thus, with the defaults, frequency translations through the 40th harmonic of 1GHz can be accurately calculated when shooting is selected as the engine.

© 1994-2020
All Rights Reserved.

When harmonic balance is selected as the engine, the number of harmonics and the oversample factor need to be set appropriately. For more information, see Chapter 3, "Frequency Domain Analyses: Harmonic Balance," Below is an example of harmonic balance in the pss *Choosing Analyses* form.

© 1994-2020
All Rights Reserved.

Regardless of the engine that is used in the pss *Choosing Analyses* form, the noise output frequency and the number of noise translations to be calculated are set up in the pnoise form. When harmonic balance is selected for the engine in pss, leave the maximum sideband field blank or set it to the number of harmonics in the pss-hb *Choosing Analyses* form, which calculates the noise translations for all the harmonics that are calculated in the pss analysis. When shooting is selected, you must specify the number of noise translations you want to calculate in the maximum sidebands field. Up to 40 noise translations can be calculated with

© 1994-2020 All Rights Reserved.

the default settings in pss. The example below shows 40 sidebands, which is the number of harmonics used in the pss-hb analysis that is shown in the preceding figure.

© 1994-2020
All Rights Reserved.

The reference sideband setting is used to specify the passband frequency for the noise figure calculation.

In the example above, the output frequency is near baseband.

The reference sideband selects the passband frequency for the noise figure calculation. Once both forms are set up, the simulation is run. To plot outputs, select *Results - Direct Plot - Main Form* in the ADE Explorer window.

The top of the form lists all the analyses that were run. Select *pnoise* where the noise figure is calculated. Next, select *NF* because that is what is desired to be plotted. If the noise figure over a bandwidth is desired, select *Integrated Over Bandwidth*, and specify the frequency range you desire for the measurement.

© 1994-2020
All Rights Reserved.

When you are done, click *Plot*.

The waveform window plots the noise figure curve and the integrated noise figure calculation over the frequency range you specified in the direct plot form. This is shown below.

Principles of Pnoise

Pnoise has three basic types of noise calculation. The first mode calculates the average noise power that occurs with the variations of the LO or clock signal taken into account. The noise is averaged over the LO or clock cycle. This is calculated when the `noisetype` parameter is set to `timeaverage`. The AM and PM components of the noise can also be calculated from `timeaverage`. For oscillators, when PM is selected, an averaged measurement can be made for FM jitter as well as phase noise. When the `noisetype` is set to `sampled(jitter)`, an instantaneous measurement of the noise at the threshold crossing in the pss analysis is calculated.This applies to both driven circuits and oscillators. There is a third setting in the *Choosing Analyses* form which is called `timedomain`. Conceptually, this does the same noise measurement as `noisetype=sampled(jitter)`, but it requires you to do the jitter calculation manually. It also allows multiple places in the rising and/or falling edges to be measured.

© 1994-2020
All Rights Reserved.

Noise Type=timeaverage

Noise Output Near Zero Frequency

First, the pss analysis is run to calculate the nonlinearity and the harmonics. Then pnoise is run to calculate the noise. In all systems, there is a maximum number of noise translations that need to be taken into account. Either the system harmonics become so small that the amount of noise translation becomes minimal or the devices themselves have finite noise bandwidths.

These noise calculations from mixing or aliasing with the different harmonics are called sidebands.

In the diagram above the X axis is the frequency. The top represents the harmonics that are calculated by the pss analysis. The bottom represents the noise output frequency. The arrows show the various mixing paths to the output frequency. Pnoise calculates the noise contribution at the output with all these mixing paths in place. Black is the noise that occurs without mixing or aliasing. Purple is the noise that mixes with the first harmonic. Blue is the noise that mixes with the second harmonic. Red is the noise that mixes with the third harmonic.

In pnoise, the output frequency range is specified along with the highest harmonic to calculate the mixing products.

© 1994-2020 All Rights Reserved.

Noise Output Near The First Harmonic

Here, the noise output frequency is just above the first harmonic frequency. Black is the noise that occurs without mixing. Purple is the noise that mixes with the first harmonic. Blue is the noise that mixes with the second harmonic. Note that the noise frequency just below the first harmonic mixes with the second harmonic to provide the output just above the first harmonic. Red is the noise that mixes with the third harmonic. Note that the second term that mixes with the third harmonic is just above the fourth harmonic of the pss analysis which is not shown. The noise term shown with a dotted line is mixing with the fourth harmonic of the system.

Frequency Sweep

Since the fundamental quantity that is calculated in pnoise is the output noise, the frequency range defined in the *Choosing Analyses* form defines an output frequency range to calculate the noise for. For driven circuits, the default is to sweep an absolute frequency range. Oscillators will be covered later in this chapter.

For an up-conversion system, sometimes, a log frequency range is desired near one of the harmonics of the system. In that case, select relative sweep, and specify a harmonic number. In the output frequency range, type the desired offset frequency from that harmonic. Below is

© 1994-2020 All Rights Reserved.

a diagram that shows relative to the first harmonic, which calculates the frequencies just above the first harmonic.

When a log sweep is desired below the first harmonic, set the relative harmonic to -1 as shown in the diagram below.

When automatic is selected for sweep type, 100 total points will be calculated. In most cases, fewer points are needed, so it is usually better to select linear or log and then specify the sweep you desire. Typically, 3 to 5 points per decade or about 20 total points are needed.

© 1994-2020 All Rights Reserved.

Maximum Sideband

There is a property in the *Choosing Analyses* form called *Maximum sideband* that is used to define how many mixing products should be calculated. If *Maximum sideband* is zero, the noise analysis will only contain the noise that appears without mixing or aliasing of any type. This is not equivalent to a linear noise analysis because pnoise takes into account the instantaneously varying nature of the noise because of the large signal being present in the pss analysis.

When *Maximum sideband* is one, then the noise without frequency translation and the noise that mixes or aliases with the first harmonic of the pss is present in the noise analysis. When *Maximum sideband* is 10, the noise analysis includes noise mixing through the 10th harmonic of the pss analysis. When harmonic balance is selected as the pss engine, the number of harmonics in the pss analysis and the maximum number of sidebands should be the same. This is easily accomplished by leaving the *Maximum sideband* field blank in the *Choosing Analyses* form in ADE Explorer. When shooting is selected for the pss engine, up to 40 sidebands can be set in the maximum sideband field by default. If you need more than 40 sidebands in pnoise, consider using full-spectrum pnoise.

Full-Spectrum Pnoise

Full-Spectrum pnoise is useful for circuits like switched-capacitor filters or sampling circuits where aliasing occurs through very high harmonics of the clock. The runtime advantages are large with no loss in accuracy of the result. Full-Spectrum pnoise is available when shooting is selected for the pss engine and APS is selected in the *Setup - High Performance Simulation* menu in ADE Explorer. Selecting *fullspectrum* in the pnoise form forces APS to be selected in ADE Explorer. If you are running from the command line without using aps, fullspectrum will not be used. In normal pnoise when shooting is selected, you need to set the maximum sideband term. In full-spectrum pnoise you do not except in cases where the pss beat frequency is 100KHz or less. In this case, set maximum sidebands to the 1/F noise corner frequency divided by the pss beat frequency. Pnoise calculates all the noise translations it can, based on the maximum timestep in the pss analysis. The easiest way to change the maximum timestep in pss is to increase the number of harmonics above 10 in the pss *Choosing Analyses* form. Full-spectrum pnoise runs faster than normal pnoise when the maximum number of sidebands in pnoise is about 50 or greater. The larger the number of sidebands, the larger the speedup for fullspectrum.

Fullspectrum pnoise is available for all available noisetypes. Noise separation is not supported when fullspectrum is selected.

© 1994-2020 All Rights Reserved.

Setting Harmonics and Sidebands

Harmonic Balance as the PSS engine

The process of setting harmonics and sidebands is similar to setting the maximum number of harmonics for the harmonic balance simulation. Start with an estimate based on the harmonic content of the input signal(s) and the nonlinearity of the circuit. Then run a simulation. Raise harmonics and sidebands and run the simulation again. If the noise measurement did not change, then the original number of harmonics and sidebands may be reduced. If it did change, then more are needed. For more details on this, refer to the section Harmonic Balance Noise Analysis (hbnoise) on page 375.

Shooting as the PSS Engine

For full-spectrum pnoise, start with 10 pss harmonics, Select full-spectrum, and select APS. Run the simulation, and plot the noise result. Now increase the number of harmonics in the pss analysis, which forces more pss timepoints, and run the simulation again. If the pnoise result did not change, then you had enough harmonics to begin with. If the noise result did change, then increase the number of pss harmonics and run the simulation again.

For regular pnoise, estimate the number of sidebands based on the pss harmonic plot. As a coarse estimate, find the harmonic where the LO or clock signal drops to about 40 dB below the amplitude of the LO or clock first harmonic. For example, if this is the 10th harmonic, set maximum sideband to 10. Run the simulation and plot the noise measurement. Increase the number of sidebands until the noise result does not change anymore. Use the smallest number of sidebands that produces a stable result. If the number of sidebands is greater than about 50, consider using full-spectrum pnoise because it is likely to run faster than normal pnoise.

Noise Separation

In addition to the total output noise, the individual noise contributions can be plotted if noise separation is selected in the *Choosing Analyses* form. More information about the noise separation will be provided later. Noise separation is available when the noise type is set to `timeaverage` and is not available for fullspectrum pnoise. The things that can be plotted are:

1. Total noise at the output from each individual noise input frequency. This allows the identification of which noise frequencies are causing the noise problem.
2. The noise at the output from each instance name with all noise mechanisms included for mixing from an individual noise input frequency. Once the noise frequency is identified in point 1 above, this capability plots the total noise from each component at the

© 1994-2020 All Rights Reserved.

troublesome noise frequency. This allows the identification of which component in the circuit is causing the problem.

3. The noise at the output from each instance name with all noise mechanisms broken out separately for mixing from an individual noise input frequency. This capability splits out the individual noise sources at the troublesome noise frequency. For example, if a parasitic resistor was found to be the problem, the component might be resized to reduce the noise.
4. The current noise at the noise instance for each instance name with all noise mechanisms included for mixing from an individual noise input frequency. This capability has limited value. For this plot, all the noise source currents for all the noise mechanisms within the component are added together.
5. The noise current in the instance from each instance name with all noise mechanisms broken out separately for mixing from an individual noise input frequency. This capability allows the identification of which noise sources are the largest in the circuit. The measurement is at the source, not at the output.
6. The gain to the output from each individual instance with all the individual noise sources broken out separately for mixing from an individual input frequency. This capability plots the gain from the noise sources in point 5 above to the output of the circuit.

Multiple Pnoise

Some systems contain multiple points in the circuit where the noise measurement is desired. When the *Noise Type* is *timeaverage*, multiple pnoise analyses are allowed. At the top of the pnoise *Choosing Analyses* form, an option called *Multiple pnoise* is provided. When this option is selected, up to six pnoise analyses can be made after a single pss analysis.

AM and PM Noise for Driven Circuits and Oscillators

Starting with IC617, the noise UI has been re-designed. Modulated has been removed and its functionalities have been merged to timeaverage, leaving three noise types now. In addition, the FM jitter option has been removed. FM jitter is calculated through timeaverage PM noise, which should be used when you want to get phase noise.

When *Noise Type* is *timeaverage*, USB, LSB, AM and PM noise are directly available. You can choose, for example, PM noise alone, or you can select the ALL(AM,PM,USB,LSB) option, if all the four types are needed. The available options are *USB*, *AM*, *PM*, *AM&PM*, and *ALL(AM,PM,USB,LSB)*.

AM and PM noise is discussed below.

© 1994-2020 All Rights Reserved.

To illustrate AM noise, see the phasor diagram below.

This diagram illustrates a periodic signal with AM noise. The black phasor is the output signal, and AM noise is a random vector that varies the amplitude of the signal.

© 1994-2020
All Rights Reserved.

PM noise is shown in the phasor diagram below.

In this case, the amplitude is constant, but the noise is a random vector perpendicular to the signal phasor. The timing of the signal varies a bit from cycle to cycle.

FM jitter and phase noise are only available through the PM noise option.

© 1994-2020 All Rights Reserved.

Most circuits produce a combination of AM and PM noise. This concept is shown below.

In this case, a random vector at any angle and amplitude is produced that varies at each instant of time. If the uncertainty creates a round shape, the AM component and the PM component are the same size.

© 1994-2020
All Rights Reserved.

In many circuits, AM or PM noise may dominate. If the PM noise dominates, the circle turns into an ellipse that is larger in the perpendicular direction than in the parallel direction. This is shown below.

Noise Type=sampled(jitter)

Jitter is usually applied to digital circuits where the average noise over the entire waveform is not appropriate. Noise in high state and low state does not matter because the output voltage is so far from the voltage that triggers the load that it does not matter. The only time the noise matters is at the exact time where the trigger event is generated. Also, the noise in the low and high state is quite small because one device is completely off, and the other device is driven on. It is basically a resistor. The device is not as good as an amplifier because the output is pulled hard into a rail. As the state changes, both devices are on, and they are relatively better amplifiers. As a result, the noise is likely to be larger in the switching interval. In jitter, the voltage for the noise measurement, and the selection of the rising edge, falling edge, or both, is made. At the specified threshold voltage, a noise measurement is made at that instant in time. Once the noise voltage is known, the timing uncertainty can be determined from the slew rate. Since the RMS noise voltage is calculated, the timing

© 1994-2020 All Rights Reserved.

uncertainty is one standard deviation from the ideal crossing time. This is illustrated below. Currently, this calculation is done in post-processing with a Direct Plot function.

sampled(jitter) has three modes. Edge Crossing is the mode just described. You specify a threshold and direction called the trigger, which defines the time when the noise measurement is made. In many cases, the measurement net will be on the same net as the trigger as shown in the diagram above, but the measurement can be made on another net in the circuit. This is useful when you want to make a noise measurement on the output of a switched-capacitor filter triggered by the clock waveform.

The Edge Delay mode is similar to edge crossing mode except that both the trigger and the measurement are edges in the circuit. For example, you might use the rising edge of a net and the falling edge for the same net to get the jitter of the pulse. Because both the trigger and the measurement are edges, the simulator calculates the jitter internally. This is done because the slew rate of the rising and falling edges are usually not the same.

The Sampled Phase mode is the equivalent of the old time-domain pnoise selection. You specify a list of PSS times you want to use to calculate the noise voltage. There is a Direct Plot function to calculate the noise voltage at the different times. If you then divide by the derivative (slope) of the PSS waveform, you get the jitter.

Noise Calculations

Phase Noise (Driven Circuit)

For a driven circuit with a single input frequency, phase noise is the total noise at the output divided by the amplitude of the first harmonic at the output node specified in the pnoise analysis. Note that phase noise from the pnoise results selection is a single-sideband phase

© 1994-2020 All Rights Reserved.

noise. Only the frequencies above the oscillator frequency are included in the measurement. To get a double-sideband phase noise measurement, set the noise type to sampled(jitter), and select the pnoise jitter result set.

Noise Factor and Noise Figure

When the pnoise analysis is set up with voltage selected instead of probe for the input, noise figure and noise factor will not have the noise of the load subtracted automatically from the measurement. Using a port as the input probe is required to obtain noise figure and noise factor calculations.

In the *Direct Plot Form*, noise factor and noise figure are both single-sideband measurements.

Noise Factor

Noise Factor is:

```
((All the noise power at the output from all the components with all the frequency
translations) - (Noise power at the output from the load))/Noise power at the output
from the resistance of the input port from the passband frequency only.
```

Note that in order to calculate the noise factor, a port component must be used as the input. In the *Choosing Analyses* form, the input port is identified by the instance name. The port component is a voltage source in series with a resistor and is considered as a single element. No access is provided to the results for the internal node where the resistor and voltage source join.

In order to determine the passband frequency, the reference sideband must be supplied. This specifies the passband frequency for the noise figure calculation. The easiest way to select the reference sideband is to use the *Select from list* feature in the pnoise *Choosing Analyses* form.

The noise from the load is normally defined by selecting the port or resistor that is serving as the load for the circuit in the *Choosing Analyses* form.

Noise Figure

Noise figure is:

```
10*log(10) (Noise Factor)
```

Noise factor (IEEE) is similar to the noise figure measurement above. If you subtract from the numerator all the noise from the input port at all frequencies except the passband, you have the IEEE noise factor.

© 1994-2020 All Rights Reserved.

Note that in order to calculate the IEEE noise factor, a port component must be used as the input. In the *Choosing Analyses* form, the input port is identified by the instance name. The port component is a voltage source in series with a resistor and is considered as a single element. No access is provided to the results for the internal node where the resistor and voltage source join.

In order to determine the passband frequency, the reference sideband must be supplied. This specifies the passband frequency for the noise figure calculation. The easiest way to select the reference sideband is to use the *Select from list* feature in the *Choosing Analyses* form.

The noise from the load is normally defined by selecting the port or resistor that is serving as the load for the circuit in the *Choosing Analyses* form.

Noise Factor IEEE

Noise Factor IEEE is:

```
((All the noise power at the output from all the components with all the frequency
translations) - (Noise power at the output from the resistor in the input port at
all frequencies except for the passband) - (Noise power at the output from the
load))/Noise power at the output from the resistance of the input port from the
passband frequency only.
```

Noise Figure IEEE

Noise figure IEEE is:

```
10*log(10) (IEEE Noise Factor)
```

Noise factor DSB is similar to the single sideband noise factor above, except in the denominator, the noise power from the input port at the image frequency is added.

Note that Noise Figure and NFiee are both single-sideband measurements. The difference between the measurements is that in noise figure, all the noise frequencies from the input port that contribute noise at the output are included, and in NFieee, only the noise at the passband

© 1994-2020 All Rights Reserved.

frequency is included. As a result, NFieee will produce a lower noise figure than Noise Figure. The difference between the measurements depends on the circuit.

.

Noise Factor DSB

Noise factor (DSB) is:

```
((All the noise power at the output from all the components with all the frequency
translations) - (Noise power at the output from the load))/Noise power at the output
from the resistance of the input port from the passband and image frequencies only.
```

Note that in order to calculate the DSB noise factor, a port component must be used as the input. In the *Choosing Analyses* form, the input port is identified by the instance name. The port component is a voltage source in series with a resistor and is considered as a single element. No access is provided to the results for the internal node where the resistor and voltage source join.

In order to determine the passband frequency, the reference sideband must be supplied. This specifies the passband frequency and by inference the image frequency for the noise figure calculation. The easiest way to select the reference sideband is to use the *Select from list* feature in the *Choosing Analyses* form.

© 1994-2020 All Rights Reserved.

The noise from the load is normally defined by selecting the port or resistor that is serving as the load for the circuit in the *Choosing Analyses* form.

Noise Figure DSB

Noise figure (DSB) is:

```
10*log(10) (Noise Factor (DSB))
```

Summary:

`No` = total output noise
`Ns` = noise at the output due to the input probe at the passband frequency (the source)
`Nsi` = noise at the output due to the image harmonic at the source
`Nso` = noise at the output due to harmonics other than input at the source
`Nl` = noise at the output due to the output probe (the load)
`IRN` = input referred noise
`G` = gain of the circuit
`F` = noise factor
`NF` = noise figure
`Fdsb` = double sideband noise factor
`NFdsb` = double sideband noise figure
`Fieee` = IEEE single sideband noise factor
`NFieee` = IEEE single sideband noise figure

then,

```
IRN = sqrt(No^2/G^2)
F = (No^2 - Nl^2)/Ns^2
NF = 10*log10(F)
Fdsb = (No^2 - Nl^2)/(Ns^2+Nsi^2)
NFdsb = 10*log10(Fdsb)
Fieee = (No^2 - Nl^2 - Nso^2)/Ns^2
NFieee = 10*log10(Fieee)
```

When the pnoise analysis is set up with voltage selected instead of probe, noise figure cannot be calculated.

Input-Referred Noise

Important

When the pnoise analysis is set up with the input source being a port, input-referred noise may not be exact. In this case, the input-referred noise is referred to the voltage source that is inside the port component. The amplitude of the noise is

© 1994-2020 All Rights Reserved.

divided by two, which assumes a perfect match. If an accurate input-referred noise is desired, set the *Input Source* to *voltage*, and select the input node. This will disable the noise figure calculation.

Noise transfer function

Pnoise calculates the forward gain from the input passband to the output and saves it. In the *Direct Plot Form*, this is called *Transfer Function*, and is in linear volts at the output divided by volts at the input.

© 1994-2020
All Rights Reserved.

Commonly Used Pnoise Options

© 1994-2020
All Rights Reserved.

Solver

This option is used for driven circuits only.

The default is *turbo*.

For harmonic balance pnoise, leave this option at the default value. For shooting pnoise, sometimes you will see warning messages that the residual is larger than the tolerance specified. This usually occurs at low offset frequencies. If you see this message, then select the *std* solver, which is more able to handle low offset frequencies. The runtime will increase when the std solver is selected.

Oscsolver

This option is used for oscillators only.

The default is *turbo*.

For harmonic balance pnoise, leave this option at the default value. For shooting pnoise, sometimes, you will see warning messages that the residual is larger than the tolerance specified. This usually occurs at low offset frequencies. If you see this message, then select the `std` solver, which is more able to handle low offset frequencies. The runtime will increase when the std solver is selected.

The `ira` solver may use less memory than turbo or std, but it is less robust for convergence.

TD Noise Algorithm Version

Leave this option at the default. (dts)

If your circuit has low frequency poles, (usually from a biasing circuit) the default dts solver is much more accurate than the standard solver.

hbprecond_solver

This option is only available when harmonic balance is set for the pss solver. When APS is not used, only the basic solver is available. When APS is used, first a much faster preconditioner is used. Occasionally, this preconditioner will stagnate, and when it does, it issues a warning that it is switching to the basic solver, and it reverts back to the basic solver. A preconditioner is a mathematical algorithm that makes the iterative matrix solving process faster.

© 1994-2020 All Rights Reserved.

Lorentzian

There are two ways of looking at phase noise. One way is to think of measuring the spectral output of the oscillator with a network analyzer. In this case, as the frequency gets close to the oscillator frequency, the amplitude starts going up, levels off, and then drops as the oscillator frequency is passed. In this case, the noise cannot be larger than the oscillations themselves, which is 0 dBc. If this is how you view phase noise, set the *lorentzian* option to *yes*.

If you think in terms of jitter, in one cycle of the oscillator output, because of the noise, you can calculate one standard deviation of the timing. If you go two cycles, and you integrate the noise with respect to time, you get twice as much noise power, or about 1.4 times the noise voltage or jitter. When you get to an infinite number of oscillator cycles, the jitter also becomes infinite. As you increase the time, you are lowering the noise frequency, and in this case, as the time becomes large, the frequency becomes very low. At infinite time, you get infinite jitter, and this occurs at zero offset from the carrier. Jitter is just another way of looking at phase noise. In other words, the phase noise can easily exceed the carrier amplitude (0 dBc) as the offset frequency becomes small because it is heading to infinity at zero frequency offset. If this is how you view phase noise, set the *lorentzian* option to *no*.

Lorentzian=yes means calculate the leveling off. Lorentzian=cornerfreqonly calculates the continuously rising phase noise, but it places a marker on the phase noise curve at the frequency where the phase noise would level off. No causes the phase noise to continue to rise as the offset frequency approaches zero.

Note: The *lorentzian* option is not available when *noisetype* is set to *sampled(jitter)* or *timedomain*.

Tolerance

Leave this option at the default.

Pnoise uses an iterative solver to calculate the noise, and any iterative solver needs a tolerance term to specify when the system has been solved accurately enough. The default for shooting pnoise is 1e-9. Harmonic balance driven circuit pnoise tolerance is 1e-6 and 1e-4 for oscillators.

Gear_Order

Do not set this option.

© 1994-2020 All Rights Reserved.

Resgmrescycle

Leave this option at the default, which is *short*.

Saveallsidebands

Leave this option at the default value of *no*. If you want to see the noise contributors from the different noise input frequencies, select *yes* for noise separation in the pnoise main *Choosing Analyses* form.

Enable Osc PPV

This option is obsolete.

AdditionalParams

This is typically used for customers who are evaluating new features. If you know the option name and the value, you can type it here. For example, you can type solver=std in this field instead of using the check box in the options form.

ADE Implementation

General Notes

All pnoise types are supported whether the pss engine is shooting or harmonic balance. Because Chapter 3 in this manual talks about harmonic balance exclusively, in this Chapter most of the examples will reference shooting pss. Where there are differences between shooting and harmonic balance, those differences will be noted. Otherwise, shooting and harmonic balance are completely interchangeable, and will provide the same answer as long as enough harmonics are used in harmonic balance.

Because shooting takes a minimum of 200 timepoints in the pss, the pss data inherently contains data through the 100th harmonic of the pss. For this reason, unless there is a requirement to calculate noise translations through more than the 40th harmonic of the pss, it is seldom required to force more harmonics in shooting pss. Also with shooting pss, fullspectrum is available, which does not require setting sidebands at all in most cases. However, if the pss beat frequency is 100KHz or less, then set maximum sideband to the 1/F noise corner frequency divided by the pss beat frequency. In harmonic balance, only the harmonics that are solved in the pss exist. When harmonic balance is selected, if you set maximum sideband larger than the number of pss harmonics, a warning message will be

© 1994-2020 All Rights Reserved.

entered in the Spectre output log, and maximum sideband will be reset to the number of pss harmonics.

Driven Circuits

Noise Type = timeaverage USB noise

On the *Choosing Analyses* form:

© 1994-2020 All Rights Reserved.

1. First, set up the pss analysis. For more information on the pss analysis, see the beginning of this chapter.

© 1994-2020
All Rights Reserved.

2. In the pnoise Choosing Analysis form, specify the output frequency range of your circuit.

Note: Since the fundamental quantity that is calculated by the noise analysis is the noise at the output, the frequency range is always the output frequency range.

3. Set about 3 to 5 points per decade in the *Points Per Decade* field.

4. When hb is selected as the engine in the pss form, leave the *Maximum sideband* field blank or set it to the same number as the number of harmonics in the HB analysis. This specifies that the noise that mixes with all the hb harmonics is calculated. When the pss engine is set to shooting, first decide whether or not to use full-spectrum pnoise, and if the default method is used, set a number of sidebands that is appropriate for your circuit. APS is selected automatically and is required when using full-spectrum pnoise. When fullspectrum pnoise is used, leave the Maximum Sideband field blank except for some special cases. If the PSS frequency is lower than the flicker noise corner frequency, set maximum sidebands to the next larger integer from the ratio of the noise corner frequency divided by the PSS frequency. In the case of an oscillator with a divider and you want to measure the flicker noise on the oscillator, set maximum sidebands to the divider ratio.

5. Specify the output of the circuit. When output is set to *probe*, a resistor or a port can be selected and this noise will be subtracted from the noise figure calculation. This is an easy way to get the correct answer for a noise figure calculation.

   The nodes that the probe is connected to also define the output nodes for the circuit. If you connect the probe across a differential circuit, the noise measurement is for that

© 1994-2020 All Rights Reserved.

differential circuit. Also note that when you select a probe, it is only used in the noise figure calculation. The output noise includes the noise of the selected element.

6. Select *port* from the *Input Source* drop-down list for noise figure calculation.
7. Choose *Select from list* from the drop-down list for reference sideband, which specifies the input frequency range (also called the passband frequency range) for the noise figure calculation.

8. Select the passband frequency range from the list.
9. Select *timeaverage* from the *Noise Type* drop-down list to calculate the average noise power at the output averaged over the entire time of one cycle of the input waveform.

© 1994-2020
All Rights Reserved.

10. Select the *Noise Separation* check box to enable plotting of the individual noise contributors in the circuit. Noise separation is only available when *Noise Type* is set to *timeaverage*.

Multiple pnoise

When *Noise type* is set to *timeaverage*, multiple pnoise analyses can be run after a single pss analysis. This is useful for LNA-Mixer combinations, or in any circuit where there are multiple points in the circuit where noise is important.

To run multiple analyses after a single pss:

© 1994-2020 All Rights Reserved.

1. First, set up the pss analysis as appropriate for your circuit. For more information, please see the pss section at the beginning of this chapter.

© 1994-2020
All Rights Reserved.

2. Select the *Multiple pnoise* check box located at the top of the pnoise *Choosing Analyses* form.

3. Set up all the noise analyses you want to run, as shown below.

© 1994-2020
All Rights Reserved.

Noise Type = timeaverage AM, PM, AM&PM noise

Timeaverage allows the measurement of the AM and PM components of noise separately.

First, set up the pss analysis as appropriate for your circuit. For more information, see the periodic steady-state section at the beginning of this chapter.

© 1994-2020
All Rights Reserved.

On the pnoise *Choosing Analyses* form:

1. Set up an appropriate frequency range of interest.

2. Set an appropriate number of sidebands. When harmonic balance is selected for the pss analysis, this field should be left blank. Fullspectrum is available for shooting only, and will be faster than the default pnoise when the number of noise sidebands is about 50 or greater with no loss in accuracy.

3. Select *timeaverage* from the *Noise Type* drop-down list.

Run the simulation.

© 1994-2020
All Rights Reserved.

Now plot the Output noise using the Direct Plot Form.

On the *Direct Plot Form*:

1. Select *pnoise* from the *Analysis* section.
2. Select *AM* or *PM* from the *Noise Type* section. *AM* is selected here. Note that you can choose SSB or DSB for AM and PM noise, DSB is 3dB higher than SSB noise.
3. Select V/sqrt(hz) from the *Units* section.

© 1994-2020
All Rights Reserved.

4. Click *Plot*.

Noise Type=sampled(jitter)

This selection has three modes:

- Edge Crossing Mode
- Edge Delay Mode
- Sampled Phase mode

sampled(jitter) is usually used for measuring the noise in a digital circuit at a specific threshold crossing. Usually, shooting pss is used to capture the waveform, but this capability is also implemented in harmonic balance. Harmonic balance will considerably take longer than shooting PSS so it is discouraged.

First set up a PSS analysis as shown below.

© 1994-2020 All Rights Reserved.

© 1994-2020
All Rights Reserved.

An appropriate threshold is needed, which can be determined by plotting the time domain waveform from the pss Direct Plot Form. The waveforms below show an appropriate threshold value to be 1.65V. Int2 and int3 are the nets at the input and output of an inverter.

On the pnoise Choosing Analyses form,

1. Set the stop frequency as half the input frequency.
2. Set the start frequency 4 decades below that.
3. Select *Add Specific Points* and log sweep with about 5 points per decade.
4. Select the sidebands method.

   Generally, fullspectrum pnoise is recommended.
5. Leave the *maximum sidebands* field blank when using *fullspectrum*.
6. Set the *Noise Type* to *sampled(jitter)*.
7. (Optional) Specify *Sample Ratio* as half the oscillator frequency.

   Sample ratio is the ratio between sampled frequency and fundamental frequency.

© 1994-2020 All Rights Reserved.

Edge Crossing Mode

In this mode, an edge sets the time of the measurement and the measurement net specifies where in the circuit the jitter measurement should be made. In the example below, the trigger is the rising edge of the output net as it crosses VDD/2 and the noise is measured on the same net. For switched-capacitor filters, the trigger could be the clock and the measurement could be the output net of the filter

.

© 1994-2020
All Rights Reserved.

When the simulation completes, use the Direct Plot Form to plot the jitter. For driven circuits, select *Jee*, which is an edge-to-edge jitter.

In the *Modifier* section, *UI* is unit interval, and *ppm* is parts per million.

© 1994-2020
All Rights Reserved.

The output noise frequency response and the edge-to-edge jitter are displayed. The jee measurement is one standard deviation for the expected jitter.

© 1994-2020
All Rights Reserved.

Note that Edge Phase Noise is available in the *Function* section as a plot option. This is normally used for oscillators. It plots the single-sideband phase noise of the circuit.

Sampled Phase mode

This mode is equivalent to the old time-domain noise. In this mode, you specify a list of PSS times to use for noise voltage calculation. This is useful for digital circuits because you know that the threshold will vary as the process changes. This mode can give you a way of gauging how jitter varies during the transition.

First, plot a PSS waveform for the net you want to measure. Decide the range of voltages in the waveform to calculate the noise voltage. In this example, VDD/2 is 1.65V, so we place

© 1994-2020 All Rights Reserved.

markers at 0.65V and 2.65V. Now, read the PSS times from the markers. In this case, times from 72psec to 84psec are useful.

Now, set up the pnoise Choosing Analyses form. In the form:

1. Select *Sampled Phase*.
2. Select the desired net for the output.
3. Type `1`in the *Samples Per Period* field.
4. Click the *Add Specific Points* check box.
5. Type in a list of times separated by a space in the field to the right.

   In this example, there are points from 72p to 84p every one p.
6. Click *Add*.

© 1994-2020 All Rights Reserved.

Note that this setup has been added to the list. There is no limit to the number of analyses in sampled(jitter) mode.

7. Run the simulation.

After the simulation completes, select *Results - Direct Plot - Main Form*.

In the Direct Plot Form,

1. Select *pnoise sampled* from the *Analysis* section.
2. Set the *Measurement Selection* to the run with the sampled phase results.
3. Select *Integrated Output Noise* from the *Function* section.
4. If the frequency range is not set, set it to the frequency range used in the pnoise analysis.

© 1994-2020 All Rights Reserved.

5. Click *Plot*.

© 1994-2020
All Rights Reserved.

The result is shown in a new subwindow.

© 1994-2020
All Rights Reserved.

In the left subwindow, right-click on the X axis and select *Allow Any Units* from the context menu, as shown below.

© 1994-2020
All Rights Reserved.

Pull the waveform from the right subwindow into the left subwindow. Zoom in to the edge. Note the variation in noise voltage as the edge transitions.

Edge Delay Mode

In this mode, both the trigger and the measurement are edges in the circuit. To set this up,

1. Select *Edge Delay*.
2. Set the trigger edge and threshold.

   In this example, the trigger is the rising edge on the input to an inverter.
3. Set the measurement edge and threshold.

   In this example, the falling edge of the same inverter is set.
4. Click *Add*.

© 1994-2020
All Rights Reserved.

Note that a third measurement is in the list.

5. Run the simulation.

After the simulation completes click *Results - Direct Plot - Main Form*.

In the Direct Plot Form,

1. Select the edge delay run from the *Measurement Selection* drop-down list.

© 1994-2020 All Rights Reserved.

2. Select *JDelay* from the *Function* section.
3. Select *Param-Event* and select the event from the list at the right.

   In this case there is a single measurement.
4. Specify the frequency range from the *pnoise* Choosing Analyses form.

© 1994-2020 All Rights Reserved.

5. Click *Plot*.

© 1994-2020
All Rights Reserved.

The frequency response of the jitter plots and a marker is placed at the end with the jitter measurement. This is one standard deviation of the jitter.

Oscillators

Noise Type = timeaverage USB noise

The settings are the same as driven `noise type = timeaverage` with the exception of being a relative sweep by default. Since the phase noise of the oscillator is near one of the harmonics, *Sweeptype* is set to *relative* by default. A harmonic number is provided for the case where there is a frequency multiplier or a frequency divider on the output of the oscillator. Noise type = timeaverage also has the ability to plot the noise contributors from mixing with each harmonic when noise separation is selected. There is an example of noise separation later in this chapter.

The pnoise setup is similar to driven except that the relative harmonic is specified. The frequency of the relative harmonic number will be added (if the harmonic number is positive)

© 1994-2020 All Rights Reserved.

or subtracted (if the harmonic number is negative) to the frequencies in the Frequency Sweep Range field.

© 1994-2020 All Rights Reserved.

1. First, set up the pss analysis as appropriate for your circuit. For more information on the pss analysis, see the periodic steady-state analysis section at the beginning of this chapter.

© 1994-2020
All Rights Reserved.

On the pnoise *Choosing Analyses* form:

2. Specify the appropriate harmonic number for your design. In this example, the first harmonic is specified in the *Relative Harmonic* field. If you are designing an oscillator by itself, the harmonic number is one. If you have an oscillator with a frequency doubler on the output, the harmonic number is two.

3. The frequency range is added to the frequency of the harmonic number that is specified.

4. If the pss engine is harmonic balance, leave the *Maximum sideband* field blank. If the pss engine is shooting, select whether the default method or the full-spectrum method should be used. Full-spectrum is faster than the default method when more than about 50 sidebands are required for your circuit and it also does not require the maximum sideband field to be set. When the default method is selected, specify the number of desired frequency translations in the *Maximum sideband* field (shown below). This will always be specific to the design. For a rough estimate, plot the frequency domain content, and find the harmonic number where the amplitude has dropped about 40dB. Try this number in maximum sidebands. To verify that you have enough sidebands, raise the number by about 50% and run again. If the noise result did not change, then you had enough sidebands to begin with. Up to 40 sidebands can be specified by default. If you need more than 40 sidebands, set the number of pss harmonics to the number of sidebands divided by two to four.

© 1994-2020
All Rights Reserved.

5. Select *voltage* from the *Output* drop-down list.

6. Specify the output node in the *Positive Output Node* field. When the negative node is left blank, the global ground node is used.
7. Select *none* from the *Input Source* drop-down list.
8. Select *timeaverage* from the *Noise Type* drop-down list. This gives a noise measurement that is the average noise power at the output with the large signal applied in the pss analysis and the noise folding specified by the maximum sideband parameter in the *Choosing Analyses* form.

9. Select *yes* for *Noise Separation,* if you want to plot the individual noise contributors. Fullspectrum does not support noise separation.

Noise Type = timeaverage PM noise

Oscillator Phase Noise Measurement

This selection allows for measurement of cycle jitter, cycle-to-cycle jitter, and AM and PM components of phase noise. This is an averaged noise measurement over one cycle of the oscillator.

© 1994-2020 All Rights Reserved.

1. First, set up the pss form as appropriate for your circuit. For more information on the pss analysis, Chapter 4, “Overview of Periodic Steady-State (pss) Analysis,”.

© 1994-2020 All Rights Reserved.

2. On the pnoise *Choosing Analyses* form, select *timeaverage* from the *Noise Type* drop-down list.

3. Select the *ALL(AM,PM,USB,LSB)* radio button.
4. Run the simulation.

After the simulation completes *Results - Direct Plot - Main Form* in ADE Explorer.

1. Select *pnoise* from the *Analysis* section.
2. Select *Output Noise*.
3. Select dBc/Hz.
4. Select SSB (Single Sideband) or DSB. (Double Sideband)
5. Click *Plot*.

© 1994-2020 All Rights Reserved.

The result is displayed in the waveform tool.

6. Select *Jc* or *Jcc* from the Function section.

   In the Modifier section, UI is Unit interval, and ppm is parts per million.

7. Click *Add To Outputs.*

8. Click *OK*.

© 1994-2020
All Rights Reserved.

© 1994-2020
All Rights Reserved.

The Jc or Jcc calculation is displayed in the ADE window, as shown below.

Noise Type=sampled(jitter)

sampled(jitter) mode has the same three modes as for driven circuits. The most useful is the edge crossing mode for the calculation of phase noise at a threshold crossing. Note that timeaverage and sampled normally produce the same result for oscillators except at large offset frequencies. This example compares sampled and timeaverage results. Although this is supported for both shooting and harmonic balance, if the waveforms are mostly square waves, shooting will run faster and be much less error prone than harmonic balance. If you use hb, make sure that you set the number of harmonics to at least the period of the waveform divided by the fastest 10% to 90% risetime on the fastest transition in the circuit. Set up the

© 1994-2020 All Rights Reserved.

pss analysis as appropriate for your circuit. For more information, see Chapter 4, “Overview of Periodic Steady-State (pss) Analysis,”.

© 1994-2020
All Rights Reserved.

In the pnoise *Choosing Analyses* form,

1. Select sampled(*jitter)* from the *Noise Type* drop-down list.

2. Select the *Edge Crossing* mode.
3. Select the oscillator output net.
4. Specify the voltage for instantaneous noise measurement in the *Threshold Value* field.
5. Select a value from the *Crossing Direction* drop-down list.
6. Click *OK*.

© 1994-2020 All Rights Reserved.

7. Run the simulation.

After the simulation completes, select *Results* - *Direct Plot* - *Main Form* in ADE Explorer.

In the Direct Plot Form:

1. Select *pnoise sampled* from the Analysis section.
2. Select *Edge Phase Noise* from the Function section.
3. Click *Plot*.

© 1994-2020
All Rights Reserved.

The phase noise is plotted in the waveform tool, as shown below.

© 1994-2020
All Rights Reserved.

Below is a comparison of SSB phase noise and sampled(jitter) for this example. The results are the same except at large offset frequencies where the amplitudes are very small.

Note: Starting with IC6.1.7 ISR17, both *timeaveraged* and *sampled(jitter)* allow multiple pnoise analyses, however, they cannot be mixed.

Examples

Driven Circuit Noise Setup

On the *Choosing Analyses* form:

© 1994-2020 All Rights Reserved.

1. First, set up the pss analysis for your circuit. For more information see the pss section at the beginning of this chapter. Shooting is shown in the example below.

© 1994-2020 All Rights Reserved.

2. In the pnoise *Choosing Analyses* form, specify the output frequency range of your circuit.

3. If you use log spacing, set about 3 to 5 points per decade in the *Points Per Decade* field.

4. When Harmonic Balance is set for the pss engine, you can leave the *Maximum sideband* field blank. All the mixing products from the large-signal analysis will be considered. If the pss engine is shooting, select whether the default method or the full-spectrum method should be used. Full-spectrum is faster than the default method when many sidebands are required for your circuit and it does not require the sidebands to be set. When the default method is selected, specify the number of the desired frequency

© 1994-2020 All Rights Reserved.

translations in the *Maximum sideband* field. Up to 40 sidebands can be specified by default.

5. Select the resistor or port component used for the load resistance. This excludes the noise from this component for the noise figure calculation, but keeps it for the total noise output calculation.

6. If you want a noise figure calculation, you must use a port as the input for the circuit. Set the instance name.

7. The easiest way to select the reference sideband is to set it to *Select from list*.

© 1994-2020
All Rights Reserved.

8. Select the passband frequency range you want for the noise figure and input-referred noise calculations.

9. If you want to see the noise contributions from the individual frequencies and the different device noise calculations individually on the output of the noise analysis, enable noise separation. This is not supported for the full-spectrum method.

10. When you are done with the setup, run the simulation.

Noise Summary

The noise summary provides information about the devices that contribute noise to the output. It supports USB, LSB, AM and PM noise. This information is available any time a pnoise simulation completes.

In ADE Explorer, select *Results* - *Print* - *Noise Summary*. You will see the following options depending on the selection you have made for Noise Type in the pnoise Choosing Analysis form

pnoise_am - This option is available if you select the *timeaverage* option from the *NoiseType* drop-down list and select *AM* or *ALL(AM,PM,USB,LSB)* as the contribution type. *pnoise_am* has information about the amplitude modulated noise.

© 1994-2020 All Rights Reserved.

pnoise_pm - This option is available if you select the *timeaverage* option from the *NoiseType* drop-down list and select *PM* or *ALL(AM,PM,USB,LSB)* as the contribution type. *pnoise_pm* as information about the phase modulated noise.

pnoise_usb - This option is available if you select the *timeaverage* option from the *NoiseType* drop-down list and select *USB* or *ALL(AM,PM,USB,LSB)* as the contribution type. *pnoise_usb* has information about the total noise from frequencies above the output frequency.

pnoise_lsb - This option is available if you select the *timeaverage* option from the *NoiseType* drop-down list and select *ALL(AM,PM,USB,LSB)* as the contribution type. *pnoise_lsb* has information about the total noise from frequencies above the output frequency.

If noise separation is enabled, an additional *pnoise_src* option will be added along with the above options. *pnoise_src* has information about the noise currents at the individual noise sources at the noise source.

If you want the noise contributors at a single frequency, select *spot noise*. If you want to have the noise integrated over a frequency range, select *integrated noise*.

If you want noise in volts, set the *noise unit* to *V.*

You can choose a value from the *hierarchy level* drop-down box to view the noise contributions from all noise sources within each instance at the specified level of hierarchy.

Generally, *Include All Types* should be selected. If you just want the noise from specific types of noise generators, you can select them from the list.

If you want to specifically include or exclude instances in the list, use the select button to the right side of the include instances and exclude instances fields and then select the components in the schematic.

Truncate just applies to the list. The total input and output-referred noise at the bottom of the noise summary output always includes all the noise from everything.

© 1994-2020 All Rights Reserved.

The figure below is the noise summary window with the sort function being set to noise contributors. Then there are three windows that show the three different sorted outputs.

© 1994-2020
All Rights Reserved.

An example of the noise summary output sorted by noise contributors.

An example of the noise summary output sorted by composite noise.

© 1994-2020 All Rights Reserved.

The below output is sorted by device name.

```
Results Display Window
Window  Expressions  Info  Help                                                cādence

Device     % Of Total    Inp Ref Noise Param              Noise Contribution

/D0        3.02          8.33994e-08      total              4.20417e-08
                                           id                 1.07815e-08
                                           rs                 4.06357e-08
/D1        3.02          8.33994e-08      total              4.20417e-08
                                           id                 1.07815e-08
                                           rs                 4.06357e-08
/D2        3.02          8.33922e-08      total              4.2038e-08
                                           id                 1.0786e-08
                                           rs                 4.06308e-08
/D3        3.02          8.33922e-08      total              4.2038e-08
                                           id                 1.0786e-08
                                           rs                 4.06308e-08
/LO        0.00          2.12454e-23      total              1.07061e-23
                                           ext_file_noise     0
                                           rn                 1.07061e-23
/RF        18.09         2.04062e-07      total              1.02868e-07
                                           ext_file_noise     0
                                           rn                 1.02868e-07
/if        69.83         4.00966e-07      total              2.02127e-07
                                           ext_file_noise     0
                                           rn                 2.02127e-07

Integrated Noise Summary (in V) Sorted By Device Name
Total Summarized Noise = 2.41881e-07
Total Input Referred Noise = 4.81725e-07

Device     % Of Total    Inp Ref Noise Param              Noise Contribution
/if        69.83         4.00966e-07      total              2.02127e-07
                                           rn                 2.02127e-07
                                           ext_file_noise     0
/RF        18.09         2.04062e-07      total              1.02868e-07
                                           rn                 1.02868e-07
                                           ext_file_noise     0
/D0        3.02          8.33994e-08      total              4.20417e-08
                                           rs                 4.06357e-08
                                           id                 1.07815e-08
/D1        3.02          8.33994e-08      total              4.20417e-08
                                           rs                 4.06357e-08
                                           id                 1.07815e-08
/D3        3.02          8.33922e-08      total              4.2038e-08
                                           rs                 4.06308e-08
                                           id                 1.0786e-08
/D2        3.02          8.33922e-08      total              4.2038e-08
                                           rs                 4.06308e-08
                                           id                 1.0786e-08
/LO        0.00          2.12454e-23      total              1.07061e-23
                                           rn                 1.07061e-23
                                           ext_file_noise     0

Integrated Noise Summary (in V) Sorted By Device Composite Noise
Total Summarized Noise = 2.41881e-07
Total Input Referred Noise = 4.81725e-07
The above noise summary info is for pnoise_usb data
7  HelpAction
```

For all of the noise summaries, here is a table of the abbreviations used in the Param column:

© 1994-2020 All Rights Reserved.

Resistor and Inductor:

Parameter	Meaning
`rn`	resistor thermal noise
`fn`	Flicker noise

Diode:

Parameter	Meaning
`id`	Diode current shot noise
`rs`	Series parasitic resistor noise

Gummel-Poon BJT:

Parameter	Meaning
`fn`	Flicker noise
`ic`	Collector current shot noise
`ib`	Base current shot noise
`rc`	Thermal noise from the collector parasitic resistor
`rb`	Thermal noise from the base parasitic resistor
`re`	Thermal noise from the emitter parasitic resistor

© 1994-2020 All Rights Reserved.

HiCUM:

Parameter	Meaning
ib	Base current shot noise
ic	Collector current shot noise
rbi	Thermal noise from the internal base parasitic resistor (See diagram below)
rbx	Thermal noise from the external base parasitic resistor (See diagram below)
rcx	Collector parasitic resistor thermal noise (See diagram below)
re	Emitter parasitic resistor thermal noise (See diagram below)
fn	Flicker noise

Figure 4-1 HiCUM Schematic Diagram

© 1994-2020
All Rights Reserved.

VBIC:

Parameter	Meaning
rbi	Thermal noise in resistor Rbi (See diagram below)
rbx	Thermal noise in resistor Rbx (See diagram below)
rci	Thermal noise in resistor Rci (See diagram below)
re	Thermal noise in resistor Re (See diagram below)
rs	Thermal noise in resistor Rs (See diagram below)
itzf	Collector current shot noise
ibe	Total base current shot noise
ibex	Node Bx to E base current shot noise (See diagram below)
fn	Ibe flicker noise (See diagram below)
fnx	Ibex flicker noise (See diagram below)
iccp	Parasitic transport collector to emitter current noise (See diagram below)
ibep	Parasitic transport base to emitter current noise (See diagram below)
fnp	Ibep flicker Noise (See diagram below)

© 1994-2020 All Rights Reserved.

Figure 4-2 VBIC Schematic Diagram

BSIM3:

Parameter	Meaning
`fn`	Flicker Noise

© 1994-2020
All Rights Reserved.

Parameter	Meaning
id	Thermal noise from the resistive drain current
rs	Thermal noise from the source parasitic resistor
rd	Thermal noise from the drain parasitic resistor

BSIM4:

Parameter	Meaning
fn	Flicker noise
id	Thermal noise from the resistive drain current noise
rs	Source parasitic resistor thermal noise
rd	Drain parasitic resistor thermal noise
igd	Gate-to-drain tunneling current noise
igs	Gate-to-source tunneling current noise
igb	Gate-to-bulk tunneling current noise
rbdb	Resistor thermal noise between dbNode and bNode (See diagram below)
rbpb	Resistor thermal noise between bNode and bNodePrime (See diagram below)
rbpd	Resistor thermal noise between Bulk and dbNode (See diagram below)
rbps	Resistor thermal noise between bulk and bNode prime
rbsb	Resistor thermal noise between sbNode and bNode (See diagram below)
rgbi	Gate Bias-Independent Resistor thermal noise

© 1994-2020 All Rights Reserved.

Figure 4-3 BSIM4 Schematic Diagrams

PSP-103

Parameter	Meaning
`sfl`	Channel current flicker noise
`sth`	Channel current thermal noise
`sig`	induced gate noise at 1 Hz
`sigth`	Induced gate noise

© 1994-2020 All Rights Reserved.

Parameter	Meaning
`shotgd`	Gate current shot noise from gate to drain
`shotgs`	Gate current shot noise from gate to source
`sjnoise`	Source to Bulk leakage current noise
`djnoise`	Drain to Bulk leakage and avalanche current noise
`rgatenoise`	Gate resistance thermal noise
`rbulknoise`	Bulk resistor thermal noise between node BP and BI (See diagram below)
`rjnoise`	Source side bulk resistor thermal noise between node BI and BS (See diagram below)
`rjundnoise`	Drain side bulk resistor thermal noise between node BI and BD (See diagram below)
`rwellnoise`	Well resistor thermal noise between node BI and B (See diagram below)
`rsourcenoise`	Source parasitic resistor thermal noise
`rdrainnoise`	Drain parasitic resistor thermal noise

Figure 4-4 PSP-103 Schematic Diagram

© 1994-2020
All Rights Reserved.

EKV 3.0

Parameter	Meaning
`fn_id`	Channel current flicker noise
`fn_ig`	Gate current flicker noise
`rs`	Source parasitic resistor thermal noise
`rd`	Drain parasitic resistor thermal noise
`rg`	Gate parasitic resistor thermal noise
`rsb`	Thermal noise for the substrate resistor connected between BSi and B. (See diagram below)
`rdb`	Thermal noise for the substrate resistor connected between BDi and B. (See diagram below)
`rbbs`	Thermal noise for the substrate resistor connected between BSi and Bi. (See diagram below where it is labeled $R_{DSB}/2$)
`rbbd`	Thermal noise for the substrate resistor connected between BDi and Bi. (See diagram below where it is labeled $R_{DSB}/2$)
`id`	Thermal noise of the channel resistor
`cor`	Induced gate noise
`shot_ig`	Gate current shot noise

© 1994-2020 All Rights Reserved.

Figure 4-5 EKV3.0 Schematic Diagrams

For other model types:

1. In a unix/linux shell window type `which spectre`.

   This should return a path that ends in `tools/bin/spectre`.

© 1994-2020 All Rights Reserved.

2. Triple-click the line with the path, and then click the center mouse button to enter it as a command.
3. Use the backspace key, and delete the word `spectre`.
4. Type `cdnshelp` at the end of the expression, and press `Enter`.

   A browser window will be displayed.
5. Close the *Tip of the day* window.
6. At the top, you should see the release number of the simulator you are using.
7. Click the plus sign (+) to the left of the MMSim folder.
8. Click the plus sign (+) to the left of the *Spectre Circuit Simulator Components and Device Models Reference* folder.
9. Select the device type you are using.
10. At the bottom of that device listing, you will see *Component Statements*. Double-click on this.
11. Enter the name of the parameter in the search box in the right pane of the cdnshelp window.
12. Click the down arrow to the right of the *Find* field until you see the parameter name in the pane on the right side, and then see the definition of that parameter.

© 1994-2020 All Rights Reserved.

Noise Figure

On the *Direct Plot Form*:

1. Select *pnoise* from the *Analysis* section.
2. Select the desired type of noise plot from the *Function* section. *Noise Figure* is single-sideband noise figure.
3. If you want the noise figure over a specified bandwidth, check the *Integrated Over Bandwidth* checkbox.

© 1994-2020 All Rights Reserved.

4. Specify the frequency range you want for the noise figure calculation.
5. Click *Plot*.

   The dsb noise figure and the noise figure calculation over your bandwidth are displayed.

Viewing Noise Separation Results

Viewing the Total Noise at the Output from a Single Noise Input Frequency

This capability allows the noise from all sources in the circuit at different noise input frequencies to be measured. For example, all the noise that does not change frequency (the zero sideband) can be plotted, along with the other noise input frequencies (sidebands) This plotting capability allows problem frequencies to be identified.

© 1994-2020 All Rights Reserved.

On the Direct Plot Form:

© 1994-2020
All Rights Reserved.

1. Select *pnoise separation* from the *Analysis* section.
2. Select the noise type from the *Noise Type* section. By default, *USB* is selected.
3. To view the total noise at the output select *Sideband Output* from the *Function* section.
4. Select *V/sqrt(Hz)* or *v**2/Hz* from the *Signal Level* radio buttons.
5. Select the appropriate modifier from the *Modifier* section.
6. Select the sideband number form the *Output Sideband* list. 0 means no frequency translation.
7. Click *Plot*.

The output noise from sideband zero is plotted. This is the noise that is present at the output with no frequency translations included.

© 1994-2020
All Rights Reserved.

Plotting Noise from Sideband -1

On the *Direct Plot Form*:

1. Select the -1 sideband from the *Output Sideband* list. Selecting the -1 sideband means the noise frequency is shifted down in frequency by -1* first harmonic of pss from the frequency that is specified as the noise output frequency in the *Choosing Analyses* form. For example, if the pss frequency is 1.9G, and the noise output frequency is 1M, the noise input frequency is -1.899GHz.

© 1994-2020
All Rights Reserved.

2. Click *Plot*.

As a suggestion, use the *Graph - Add Label* menu option to place labels to identify the traces, as shown in the figure above.

© 1994-2020
All Rights Reserved.

If you select multiple sidebands to be plotted at the same time, the sideband number will be shown at the right side of the legend in the waveform window.

© 1994-2020
All Rights Reserved.

Viewing Total Noise at the Output from each Instance

Once the problem frequencies are identified in the plot of Sideband Output, the problem components need to be identified. From the previous plot, the zero sideband contributes the largest amount of noise to the overall output noise. Using the Instance Output allows the identification of which components in the zero sideband contribute the most noise. The total noise for all the noise mechanisms within the component is plotted.

© 1994-2020
All Rights Reserved.

On the *Direct Plot Form*:

© 1994-2020
All Rights Reserved.

1. Select *Instance Output* from the *Function* section.
2. Select *V/sqrt(Hz)* or *V**2/Hz*.
3. Select the appropriate modifier.
4. Select the sideband number. In this example, the zero sideband was determined to contribute the most noise at the output of the circuit.
5. Select the type of component to display. If you want all of the types, choose *Include All Types*.
6. Specify the number of results you want to plot in the *Truncate* section.
7. Click *Plot*.

   The total noise at the output for the top three components is plotted.

© 1994-2020
All Rights Reserved.

To see which component goes with which trace, select the plus sign (highlighted in red on the previous graphic) to the left or the text at the upper left of the display area. The legends expand, and you can see which trace is which in the Name field on the left side of the display tool. You can also either select the trace or the individual entry in the Name field and the trace and legend will highlight. The instance name is shown on the right side of the legend. You may need to change the size of the legend area by moving your mouse cursor over the line that separates the two areas in the waveform tool. The cursor will change shape. Now click-and-hold the mouse button, move to a new location, and release the mouse button.

Viewing the Exact Noise Mechanism Within the devices

The sideband with the largest contribution was determined by selecting Sideband Output and plotting. The specific component was identified by selecting Instance Output. The specific noise mechanisms can be identified using Source output. This plots the largest individual noise mechanisms for the sideband you select.

© 1994-2020 All Rights Reserved.

On the *Direct Plot Form*:

© 1994-2020
All Rights Reserved.

1. Select *Source Output* from the *Function* section.
2. Select *V/sqrt(Hz)* or *v**2/Hz*.
3. Select the appropriate modifier.
4. Select the sideband number.
5. Select the type of component to display. If you want all of the types, choose *Include All Types*.
6. If you want to restrict to one device, put its instance name in the *include inst* field. A list of devices separated by a space is also acceptable.
7. If you want to exclude a device (or devices) enter the instance name(s) in the *exclude inst* field.
8. Specify the number of results you want to plot in the *by top* field.
9. Click *Plot.*

   The exact noise mechanism within the devices is plotted.

© 1994-2020
All Rights Reserved.

To see which noise source goes with which trace, select the plus sign to the left or the text at the upper left of the display area. The legends will be shown individually in the Name field of the waveform tool. The device name and the individual noise contributor will be shown. You can select either the individual legend or the trace and both the legend and the trace will be highlighted. The first part is the instance name, and the second part is the specific noise mechanism within that instance. For a list of the abbreviations used for the noise parameter, see the list that starts on page 123 (77 of 234) In the example below, fn is flicker noise, and id is the channel resistance thermal noise.

© 1994-2020
All Rights Reserved.

Viewing the total noise currents within the devices

© 1994-2020
All Rights Reserved.

This capability has limited value. All the noise currents from all the noise mechanisms within the instance are added up. No gain function is applied or are available for this measurement. It just adds up everything within that instance.

1. Select *Instance source*.
2. Select *A/sqrt(Hz)* or *A**2/Hz* in the Signal Level selection.
3. Select *Magnitude* or *dB20* in the *Modifier* section.
4. Select the sideband number.
5. If you want every noise source to be considered, select I*nclude All Types*.
6. To exclude instances, click *Select* to the right of the *exclude inst* field, and select the device instances you want to exclude from the plot.
7. To include specific instances, click *Select* to the right of the *include inst* field, and select the instances on the schematic.
8. Set the number of curves to plot by setting the Truncate function.
9. Click *Plot*.
10. The frequency response curves are displayed.

© 1994-2020
All Rights Reserved.

Plotting the individual noise source currents

© 1994-2020
All Rights Reserved.

This capability plots the individual noise currents within the device. This capability gives you the noise at the source instead of at the output of the simulation.

1. Select *Primary Source*.
2. Select *A/sqrt(Hz)* or *A**2/Hz* in the *Signal Level* selection.
3. Select *Magnitude* or *dB20* in the Modifier section.
4. Select the sideband number.
5. If you want every noise source to be considered, select I*nclude All Types*.
6. To exclude instances, click *Select* to the right of the *exclude inst* field, and select the device instances you want to exclude from the plot.
7. To include specific instances, click *Select* to the right of the *include inst.* field, and select the instances on the schematic.
8. Set the number of curves to plot by setting the Truncate function.
9. Click *Plot*.
10. The frequency response curves are displayed.

For a list of the noise parameter names, please see the Noise Summary on page 1001.

© 1994-2020
All Rights Reserved.

Plotting the gain from the individual noise sources to the output

© 1994-2020
All Rights Reserved.

This capability gives the gain from the individual noise currents to the output of the circuit.

1. Select *Source Noise Gain*.
2. Select *A/sqrt(Hz)* or *A**2/Hz* in the *Signal Level* selection.
3. Select *Magnitude* or *dB20* in the *Modifier* section.
4. Select the sideband number.
5. If you want every noise source to be considered, select *Include All Types*.
6. To exclude instances, click *Select* to the right of the *exclude inst* field, and select the device instances you want to exclude from the plot.
7. To include specific instances, click *Select* to the right of the *include inst.* field, and select the instances on the schematic.
8. Set the number of curves to plot by setting the Truncate function.
9. Click *Plot*.

The frequency response curves are displayed. For a list of the abbreviations used for the noise parameters, please see Noise Summary on page 1001.

© 1994-2020
All Rights Reserved.

Multiple pnoise

Consider an LNA-Mixer circuit shown below.

In the above circuit, there are two outputs that need to be measured. The first is at the LNA output, and the second is at the mixer output. The noise frequencies are different at the two outputs.

© 1994-2020
All Rights Reserved.

1. First, set up the pss form as appropriate for your design. For more information on the pss analysis, please see the periodic steady-state analysis at the beginning of this chapter.

© 1994-2020 All Rights Reserved.

2. In the pnoise *Choosing Analyses* form, select *Multiple pnoise*, and set up the noise analysis for one of the outputs. The mixer output is shown below.

© 1994-2020
All Rights Reserved.

3. Select the second line in the *Multiple pnoise* section and set up that analysis.

© 1994-2020
All Rights Reserved.

4. Run the simulation. When the simulation completes, select *Results - Direct plot - Main Form*. The single-sideband noise figure measurement for the LNA is shown below. The selection mechanism for the noise result is called Multiple Out and is highlighted below.

© 1994-2020
All Rights Reserved.

5. Click *Plot*.

© 1994-2020
All Rights Reserved.

6. To plot the noise figure of the overall circuit, set the *Multiple Out* selection to the Mixer output at Port2. For direct conversion, usually double-sideband noise figure is desired.

7. Click *Plot*.

© 1994-2020
All Rights Reserved.

The waveform tool shows the noise figure curve and the integrated noise figure to 1MHz.

© 1994-2020
All Rights Reserved.

Noise as a Function of Blocker Power

- Note that the input source has 2 large-signal tones set up.

© 1994-2020
All Rights Reserved.

- The first large-signal tone has the frequency set to the variable *fb1* and the amplitude set to *pb*.
- The second tone has the frequency set to the variable *fb2* and the amplitude set to *pb*. This is the blocker signal.
- Note that for this measurement, you might need to switch to harmonic balance. If the blocker frequency and LO frequency have a low PSS beat frequency, then PSS will take a very long time to run and will use a lot of memory. In the case shown below, the PSS frequency of 100MHz is high enough to be possible.

To measure the noise as a function of blocker power:

1. Apply the signals that cause the large-signal effects in the pss analysis, and then run a pnoise analysis. In this case, the LO and the blocking signal should be applied in the pss analysis. Note that if the highest input frequency divided by the PSS beat frequency is above about 25, it is very likely that hb or qpss would run faster than pss because hb and qpss only calculate the actual frequencies produced by the circuit, while pss calculates all the harmonics of the beat frequency.
   - The LO is set to 2.4GHz (shown in green).
   - The first RF is set to 2.5G and -24 dBm (shown in red). This is the blocker signal.

© 1994-2020 All Rights Reserved.

- The second RF tone is disabled by setting its frequency to zero (shown in yellow).

© 1994-2020 All Rights Reserved.

On the pss *Choosing Analyses* form:

© 1994-2020
All Rights Reserved.

1. Set up the pss analysis appropriately. For more information, see the pss section at the beginning of the chapter.
2. Sweep the RF power over an appropriate range. Generally speaking, the power should start about 20 to 40 dB below the below the 1dB gain compression point, and sweep through the power level that is needed to meet the specification for the communication system you are working on. In general, this higher level would be near the compression point, or perhaps slightly above it.

© 1994-2020 All Rights Reserved.

On the pnoise *Choosing Analyses* form:

© 1994-2020
All Rights Reserved.

1. Usually, a single frequency is run in the pnoise, when the blocker power is swept.
2. Specify the desired output frequency.

   Shooting: Because to go through the fifth harmonic of 2.5GHz requires that sidebands be set to 125, (5*25th harmonic of 100MHz), fullspectrum pnoise is chosen so the simulation time is reduced. When fullspectrum is chosen, the number of frequency translations depends on the maximum step in the pss analysis. The maximum step of the pss analysis depends on how the maximum harmonic field is set. In this case, to verify that enough sidebands are being used, first set the maximum harmonics term to a bit less than the number anticipated for pnoise. Run the simulation, and plot the noise figure. Roughly double the number of pss harmonics, and run the simulation again. If the noise figure plot did not change, you had enough to start with, and try reducing the number of harmonics in the pss analysis. Reducing the number of harmonics below 10 will have no effect on the simulation. If the noise figure increased, then double the number of harmonics again and re-run the simulation. Use the smallest number of harmonics that produces a stable pnoise result. If the default pnoise method is used, start with an estimate of the number of frequency translations you need based on the nonlinearity of the system. Set that as the number of sidebands, Divide this number by 4, and set that as the number of pss harmonics. Run the simulation, and plot the noise result. Now double the number of pss harmonics and the number of pnoise sidebands. Rerun the simulation and replot the noise result. If the noise result did not change, you had enough harmonics and sidebands to begin with. If it did change, double again, and replot. Use the smallest number of harmonics and sidebands that give a stable result.

   Harmonic Balance: Leave the *Maximum sideband* field blank, and estimate the number of required harmonics in the pss analysis. Run the simulation and plot the noise. Now increase the number of harmonics in pss by about 50%, and run the simulation again. If the noise changed, increase the pss harmonics and run again. If the noise did not change, reduce the number of pss harmonics an run again. Use the smallest number of harmonics you can in pss that maintains the stable noise output in pnoise.
3. The rest are set as the preceding section.
4. Run the simulation when you are done.

© 1994-2020 All Rights Reserved.

5. When the simulation completes, select *Results - Direct Plot - Main Form* in ADE Explorer.

6. Select *pnoise*.
7. Select the desired noise figure measurement.
8. Click *Plot*.

© 1994-2020
All Rights Reserved.

The resulting plot appears.

9. To make the plot more useful, right-click one of the numbers on the X Axis and select *Swap Sweep Var* from the context menu.

© 1994-2020
All Rights Reserved.

The *Swap Sweep Var* window is displayed. Note that the *Sweep Variables* field is set to *prf*, which is the variable that was swept.

10. Specify 1000000 in the X Value field.

11. Click *OK*.

A new subwindow is displayed with the noise figure on the Y axis, and the blocker power on the X Axis.

© 1994-2020
All Rights Reserved.

Noise Type = timeaverage AM, PM AM&PM noise

timeaverage allows the measurement of the AM and PM components of noise separately. This can be applied to any driven circuit or oscillator. This section is for driven circuits. See the oscillator section if you have an oscillator. Note that this is an averaged measurement over the full cycle of the pss waveform, not an instantaneous measurement at a single threshold voltage.

© 1994-2020 All Rights Reserved.

First, set up the pss analysis as appropriate for your circuit. For more information, refer to the section Overview of Periodic Steady-State (pss) Analysis on page 647.

© 1994-2020
All Rights Reserved.

On the pnoise *Choosing Analyses* form:

1. Set up an appropriate frequency range of interest.

2. **Shooting**: Set maximum sideband up to 4 times the number of pss harmonics. Up to 40 sidebands can be set using the defaults. Fullspectrum is not available for harmonic balance.

Harmonic balance: Leave the *Maximum sideband* field blank. This calculates noise translations for all the harmonics in the pss-harmonic balance analysis.

3. Select the output net or nets in the schematic. If the *Negative Output Node* field is left blank, the global ground net is used as the entry.

© 1994-2020
All Rights Reserved.

4. Select *timeaverage* from the *Noise Type* drop-down list.

5. Run the simulation.

Now plot the output noise using the *Direct Plot Form*.

On the *Direct Plot Form*:

1. Select pnoise from the *Analysis* section.

© 1994-2020 All Rights Reserved.

2. Select *Output Noise* from the *Function* section.

3. Enter the frequency in the *Value At (Hz)* field if you want to plot the value at a specific frequency. You can enter a list of frequencies separate by a space in this field.

4. If you want this signal to plot automatically when a new simulation is run, select *Add To Outputs*. The expression is automatically evaluated in ADE.

5. Click *Plot*.

© 1994-2020 All Rights Reserved.

6. Again, select *pnoise* from the *Analysis* section.
7. Select *AM* or *PM* from the *Noise Type* section. The AM noise is selected below. Note you can choose SSB or DSB noise for AM and PM, DSB is 3dB higher than SSB noise.

© 1994-2020
All Rights Reserved.

8. Click *Plot*.

On the plot, note that DSB AM noise is 3dB larger than the output noise. This is because the output noise is derived the single sideband only (the positive frequencies), and the DSB AM or PM noise is derived using the double sideband.

Noise Type=Sampled(jitter)

Driven jitter is usually used for measuring the noise in a digital circuit at a specific threshold crossing.This is an instantaneous measurement at one or more points in the time-domain waveform at a single threshold voltage. Accordingly, the pss *Choose Analyses* form uses shooting, as shown below. For more information on the pss analysis, please see Overview of Periodic Steady-State (pss) Analysis on page 647. This is also supported for harmonic balance. Harmonic balance is not recommended because the runtime is likely to be much longer than shooting. If you must use harmonic balance, measure the 10% to 90% risetimes on the rising and falling edge of the net to be measured from the tstab waveform. Set at least the period divided by the fastest risetime for the number of harmonics. If you underspecify the

© 1994-2020 All Rights Reserved.

number of harmonics in hb, the risetime of the waveform will be smaller than the real circuit, and this will reduce the accuracy of the measurement.

© 1994-2020
All Rights Reserved.

An appropriate threshold is needed, which can be determined by plotting the time domain waveform from the pss *Direct Plot Form*. The waveform below shows an appropriate threshold value to be 1.65V for this circuit, which is a 3.3 Volt logic circuit.

On the pnoise *Choosing Analyses* form:

1. The stop frequency is half the input frequency.
2. The start frequency is 4 decades below that.
3. About 5 points per decade is usually sufficient.
4. It is recommended to use fullspectrum pnoise because it will run faster.
5. *Maximum sideband* should be left blank.
6. The *Output* field is set to *voltage*.
7. The *Noise Type* is set to *sampled(jitter)*.
8. Select the *Edge Crossing* Mode.
9. Set Timing Event to *Edge Crossing*.
10. Select the net that will be used to get the time of the measurement in the *Trigger* section.

© 1994-2020 All Rights Reserved.

11. Set an appropriate threshold and edge direction as per your requirements.
12. Specify the net where the noise measurement will be made.
13. Click *OK*.

© 1994-2020
All Rights Reserved.

Now run the simulation. When the simulation completes, select *Results - Direct Plot - Main Form*.

On the Direct Plot Form, select *Jee*, which is edge-to-edge jitter. In the below example, 43p is the rising edge.

© 1994-2020
All Rights Reserved.

The output noise frequency response and the edge-to-edge jitter are displayed. The jee measurement is one standard deviation for the expected jitter.

Edge Delay Mode

In this mode, the trigger and the measurement are both edges. In the example below, the *int2* net is the input to an inverter and *int3* is the output of the inverter. In this example, the jitter of the inverter connected between *int2* and *int3* is measured. You can also specify the same net from rising to falling edge to measure the jitter of a pulse. Both trigger and measurement can be any edge in the circuit. As a result, this mode is very flexible.

In this mode, use the same frequency range as before. The stop frequency is half the PSS frequency and the start frequency is 4 decades below that.

- Select *Edge Delay* for Timing Event.
- Specify the trigger net and threshold.
- Specify the measurement net and threshold.

© 1994-2020 All Rights Reserved.

- Click *OK*.

.

- Run the simulation.

After the simulation completes, select *Results - Direct Plot - Main Form*.

On the Direct Plot Form:

© 1994-2020 All Rights Reserved.

1. Select *jDelay* from the Function section.
2. Select *rms* for Signal Level.
3. Select *Second* from the Modifier section.
4. Set the frequency range equal to the pnoise frequency range.
5. Click *Plot*.

© 1994-2020
All Rights Reserved.

Jitter is displayed in the marker.

Sampled Phase mode

This is used to measure the output noise voltage at several timepoints in the pss waveform. First set up the PSS Choosing Analyses form and plot the waveform for the desired net.

Position the markers at interesting timepoints, as shown below.

To plot this waveform, select *Results* - *Direct Plot* - *Main Form* in ADE Explorer.

1. Select *pss* from the Analysis section.
2. Select *time* from the Sweep section.
3. Select the desired net from the schematic.
4. Move your mouse cursor to the point in the waveform where you want a marker, and type `m`.

© 1994-2020 All Rights Reserved.

5. You can move the markers by moving to the dot that highlights the position of the marker on the trace, click and hold the mouse button, move to the desired location, and release the mouse button.

On the pnoise *Choosing Analyses* form:

1. Set the *Sweeptype* to *default*.
2. Set the *Stop* frequency to one half the PSS frequency.
3. Set the *Start* frequency 4 decades below the *Stop* frequency.
4. Select *Sweep Type* as *Logarithimic*.
5. Select *Points per Decade* and specify about 5 points per decade.
6. Select *Sampled Phase* for Timing Event.
7. Select the net of interest.
8. Type 1 in the *Samples Per Period* field. The default in sampled phase mode is to run a pnoise analysis at every timepoint in the pss analysis. Because noise is usually only

© 1994-2020 All Rights Reserved.

required at several timepoints, setting the number of points to 1 causes a single point at the first timepoint in the pss analysis to be run in pnoise.

9. Select the *Add Specific Points* check box and enter the time values from the markers positioned in the waveform. Use the space character to separate the entries.

10. Click *OK* and run the simulation.

© 1994-2020 All Rights Reserved.

© 1994-2020
All Rights Reserved.

After the simulation completes, select *Results - Direct Plot - Main Form*.

On the Direct Plot Form:

1. Select *pnoise sampled* from the *Analysis* section.
2. Select *Integ Output Noise* from the *Function* section.
3. Select *Add to Outputs*.
4. Click *Plot*.

© 1994-2020
All Rights Reserved.

The noise voltage is plotted as shown below. As expected, the maximum noise occurs near the 1.65 volt level of the waveform. This is shown in the right subwindow below.

© 1994-2020
All Rights Reserved.

To pull the noise waveform into the left subwindow, right-click on the X axis of the left subwindow and select *Allow Any Units* from the context menu.

- ❑ Drag the noise waveform into the left subwindow.
- ❑ Delete the right subwindow.
- ❑ Click the *Split all strips* icon.

© 1994-2020 All Rights Reserved.

Now you can see the noise voltage at different times of the edge.

© 1994-2020
All Rights Reserved.

Oscillators

Noise Type = timeaverage

Noise type=timeaverage is used to measure the single sideband phase noise that is averaged over the entire cycle of the oscillator waveform.

© 1994-2020 All Rights Reserved.

1. First, set up the pss form as shown in the following figure. For more information see the pss section at the beginning of this chapter of the user guide.

© 1994-2020
All Rights Reserved.

2. Next, set up the pnoise form as shown below. For the oscillator, you may need to change the start and stop frequencies and the output node. Select *Timeaverage* from the *Noise Type* drop-down list to measure the average noise of the oscillator.

© 1994-2020
All Rights Reserved.

The sweep defaults to relative to the frequency of the specified harmonic of the pss analysis. The frequencies specified in the *Choosing Analyses* form are added to the frequency of the specified harmonic when *relative* is selected. In this case, the first harmonic is specified because there is no frequency multiplier on the output of the oscillator.

1. Set the relative harmonic number. For an oscillator by itself, the relative harmonic is one. For an oscillator with a prescaler and you desire the output of the prescaler, the harmonic number is one. If you want the noise of the VCO without the prescaler, then the harmonic number is the divide ratio. If you have an oscillator with a frequency multiplier and you want the noise at the output of the frequency multiplier, enter the frequency multiplication ratio. If you want the noise at the output of the VCO by itself, enter one.
2. Relative to the output frequency, specify a frequency offset.
3. Generally, three to five points per decade in a log sweep are sufficient.
4. Setting sidebands depends on whether shooting or harmonic balance has been chosen as the pss engine.
   a. When harmonic balance is selected, leave the maximum sideband field blank, which causes the noise translations for all the pss harmonics to be calculated. Fullspectrum is not available for harmonic balance. Experiment with the number of harmonics and oversamplefactor to find the most efficient pss-pnoise runtime. Overview of Periodic Steady-State (pss) Analysis on page 647 for more information on setting harmonics.
   b. When shooting is the pss engine, fullspectrum is available and recommended. Fullspectrum calculates the number of noise translations to include based on the maximum timestep in the pss analysis. By default, frequency translations through the 20th harmonic are taken into account by fullspectrum pnoise. In the unusual case where the noise calculation needs to go through a very high harmonic of the pss analysis, increase the number of harmonics in the pss form to the minimum number of harmonics that causes the noise result to not change with a larger number of pss harmonics. Generally speaking, these will be switched-capacitor circuits or sampling circuits where aliasing can go through a very high harmonic of the clock. Fullspectrum on this type of circuit will run much faster than the default pnoise simulation. The larger the number of sidebands, the larger the speed differential.
5. Note that when *fullspectrum* pnoise is chosen, APS will be enabled in ADE Explorer. Full-spectrum will only be run in APS. It is not available in standard Spectre. APS requires additional tokens or different licensing to run compared to standard Spectre.
6. Set *Output* to *voltage*.
7. Select the output node.

© 1994-2020 All Rights Reserved.

8. Set *Input Sources* to *none*. Input-referred noise is not defined for an oscillator.
9. If you want noise separation, you must first select *default* for *Method* and set an appropriate number of sidebands in the *Maximum sideband* field. Then you can select the noise separation. Noise separation allows plotting of where specifically the noise is coming from.
10. Run the simulation.

After the simulation completes, select *Results* - *Direct Plot* - *Main Form* in the ADE Explorer window.

On the *Direct Plot Form*:

1. Select *pnoise* from the *Analysis* section.
2. Select *Output Noise* from the *Function* section.
3. Select dBc/Hz from the Units section.
4. Select SSB or DSB for single sideband or double sideband respectively.
5. If desired, type a list of frequencies separated by a space if you have phase noise specs at defined offset frequencies.
6. If you want expressions in ADE so that this setup will plot automatically after each simulation, click the *Add To Outputs* check box.

© 1994-2020 All Rights Reserved.

7. Click *Plot*.

Phase noise is plotted in the waveform tool, as shown below.

© 1994-2020
All Rights Reserved.

Noise Separation

One of the benefits of simulation is that since the noise terms are all calculated individually, we can store the data in a way to make these individual calculations available for viewing. To see the different noise separations, select *yes* for *Noise Separation* in the pnoise *Choosing Analyses* form.

The default in pnoise is to calculate all the noise translations that are specified in the *Maximum sideband* property in the *Choosing Analyses* form, add all these contributions at the output, and display that total noise term. Enabling noise separation allows you to see which components and/or which frequencies contribute the most noise at the output. Once you know where the noise is coming from, you can design a solution to the problem.

© 1994-2020 All Rights Reserved.

Noise separation is available for shooting and harmonic balance, but not for full spectrum shooting pnoise.

© 1994-2020
All Rights Reserved.

When the choice is made and the simulation is run, select *Results - Direct Plot - Main Form* in ADE Explorer.

Viewing the total noise at the output from each sideband

This allows the measurement of the noise at the output from the individual noise input frequencies in the circuit.

© 1994-2020 All Rights Reserved.

When noise separation is set to *yes* in the pnoise *Choosing Analyses* form, you can also plot subsets of the total noise in order to identify where the noise is coming from. In the *Direct Plot Form*, select the *pnoise separation* result set.

To view the total noise from mixing with different pss harmonics:

1. Select *Sideband Output*.

   Remember that the zero sideband means that the noise input is the same frequency as the noise output. Selecting the -1 sideband means that the noise that is below the output frequency that mixes with the first harmonic is plotted. -2 means that the noise frequency

© 1994-2020 All Rights Reserved.

is below the noise output frequency, and it mixes with the second harmonic of the pss analysis. +1 means that the noise frequency is above the noise output frequency, and mixes with the first harmonic of the pss analysis.

Example:

In this case, the PSS frequency is 1.9G. In the pnoise form, *relative* is selected, and the relative harmonic number is 1. This means that the output noise just above the first harmonic frequency is being analyzed. When the noise output frequency is 1MHz, the zero sideband is 1MHz above the first harmonic of the pss analysis. The -1 sideband is (-1 * 1.9GHz) + 1MHz + 1.9GHz), or 1MHz. The +1 sideband is at (+1 * 1.9G) + 1MHz + 1.9GHz, or 3.801GHz.

2. Select *V / sqrt(Hz)* or *V** 2 / Hz*.
3. Select *Magnitude* or *dB20* in the *Modifier* section.
4. Select one or more sidebands from the *Direct Plot Form*.

© 1994-2020 All Rights Reserved.

5. Click *Plot*. The results display in the waveform tool. A vertical marker has been placed so the individual results at 1MHz can be read out.

© 1994-2020
All Rights Reserved.

6. To see the individual contributions, select the legend at the upper left for the sideband number you want to see. To see the noise at the output with no frequency translation, select the zero sideband as shown below. The legend and the curve will highlight.

7. To see the noise that is upconverted from near zero frequency, select the -1 sideband. Selecting the -1 sideband means that the noise frequency is one times the pss frequency

© 1994-2020 All Rights Reserved.

(the oscillator frequency) below the noise output frequency. This includes the 1/F noise that upconverts to the output.

© 1994-2020
All Rights Reserved.

8. To see the noise that appears by mixing with the second harmonic, select the -2 sideband.

© 1994-2020
All Rights Reserved.

9. To see the noise converted by mixing with the third harmonic, select the -3 sideband.

Viewing the total noise at the output for all the noise mechanisms in each noise device for one noise sideband

This capability allows you to plot the total noise from each device at each separate input frequency. This gives you an idea of not only the frequencies that are troublesome, but also the specific devices that are troublesome.

© 1994-2020 All Rights Reserved.

1. Select Instance Output.

2. Select *V / sqrt(Hz)* or *V**2 / Hz*.

© 1994-2020
All Rights Reserved.

3. Select linear *Magnitude*, or *dB20*.
4. Select only one noise sideband.
5. To plot the noise from all the different device types, select *Include All Types*.
6. To make individual selections, choose each individual device type in the list. To select more than one device type at a time, press and hold the <Ctrl> key while you choose an additional category.
7. To include only the devices you select from the schematic included, select I*nclude None*, and click the *Select* button to the right of *Include Inst*. Now select the instances you want to include from the schematic. As you select the instances, the instance name will appear in the list. Select to the right of *Include inst*. Now select the instances you want in the schematic. As you click the instances, the instance Name will appear in the list.
8. In the Truncate field, type in the number of traces you want to view. The instances with the largest noise contributions will be plotted.

The highlighted trace above is a plot of all the noise contributors in *Q0* at the output of the circuit.

© 1994-2020 All Rights Reserved.

To plot the individual noise contributors form each instance from a single noise sideband

This plot splits the noise into each specific noise generator at each individual noise input frequency.

© 1994-2020 All Rights Reserved.

1. Select Source Output.

2. Select *V / sqrt(Hz)* or *V**2 / Hz*.

© 1994-2020
All Rights Reserved.

3. Select linear *Magnitude*, or *dB20*.
4. Select only one noise sideband.
5. To plot the noise from all the different device types, select *Include All Types*.
6. To make individual selections, choose each individual device type in the list. To select more than one device type at a time, press and hold the <Ctrl> key while you choose an additional category.
7. To include only the devices you select from the schematic included, select *Include None*, and click the *Select* button to the right of *Include Inst*. Now select the instances you want to include from the schematic. As you select the instances, the instance name will appear in the list. Select to the right of *Include inst*. Now select the instances you want in the schematic. As you click the instances, the instance Name will appear in the list.
8. In the Truncate field, type in the number of traces you want to view. The instances with the largest noise contributions will be plotted.

The instance name and the noise parameter are displayed in the waveform tool. See the red square in the above figure. For a table of the noise parameters, see Noise Summary on page 1001.

© 1994-2020
All Rights Reserved.

Viewing the total noise current from each instance at the source of the noise

Instance source changes the focus from the output to the noise source itself, and plots the actual noise currents at the source for each individual noise frequency. All the noise from all the individual noise sources combined for each device is available.

© 1994-2020 All Rights Reserved.

1. In the *Direct Plot Form*, select *Instance Source*.

2. Select *V / sqrt(Hz)* or *V**2 / Hz*.

© 1994-2020
All Rights Reserved.

3. Select linear *Magnitude*, or *dB20*.
4. Select only one noise sideband.
5. To plot the noise from all the different device types, select *Include All Types*.
6. To make individual selections, choose each individual device type in the list. To select more than one device type at a time, press and hold the Control key while you choose an additional category.
7. To include only the devices you select from the schematic included, select *Include None*, and click the *Select* button to the right of *Include Inst*. Now select the instances you want to include from the schematic. As you select the instances, the instance name will appear in the list. Select to the right of *Include inst*. Now select the instances you want in the schematic. As you click the instances, the instance Name will appear in the list.
8. In the *Truncate* field, type in the number of traces you want to view. The instances with the largest noise contributions will be plotted. The total noise from all sources in Q0 at 1MHz offset is about -201 picoAmps per root hertz.

Viewing the individual noise current from each noise source within each instance at the source of the noise

This splits out the individual noise currents at the source for each individual noise source in the circuit.

© 1994-2020 All Rights Reserved.

1. In the *Direct Plot Form*, select *Primary Source*.

2. Select *V / sqrt(Hz)* or *V**2 / Hz*.

© 1994-2020
All Rights Reserved.

3. Select linear *Magnitude*, or *dB20*.
4. Select only one noise sideband.
5. To plot the noise from all the different device types, select *Include All Types*.
6. To make individual selections, choose each individual device type in the list. To select more than one device type at a time, press and hold the Control key while you choose an additional category.
7. To include only the devices you select from the schematic included, select *Include None*, and click the *Select* button to the right of Include Inst. Now select the instances you want to include from the schematic. As you select the instances, the instance name will appear in the list. Select to the right of *Include inst*. Now select the instances you want in the schematic. As you click the instances, the instance Name will appear in the list.
8. In the *Truncate* field, type in the number of traces you want to view. The instances with the largest noise contributions will be plotted. The individual noise from all the different sources in Q0 at 1MHz offset is shown below. To compare the noise with the total noise shown in the previous section, add the noise power (not the voltage) from the individual noise sources by taking the square root of the sum of the squares.

© 1994-2020
All Rights Reserved.

Viewing the transfer function from the individual sources of the noise to the output of the circuit

1. In the *Direct Plot Form*, select *Src Noise Gain*.

© 1994-2020 All Rights Reserved.

2. Select *V / sqrt(Hz)* or *V**2 / Hz*.

© 1994-2020
All Rights Reserved.

3. Select linear *Magnitude*, or *dB20*.
4. Select only one noise sideband.
5. To plot the noise from all the different device types, select *Include All Types*.
6. To make individual selections, choose each individual device type in the list. To select more than one device type at a time, press and hold the Control key while you choose an additional category.
7. To include only the devices you select from the schematic included, select *Include None*, and click the *Select* button to the right of *Include Inst*. Now select the instances you want to include from the schematic. As you select the instances, the instance name will appear in the list. Select to the right of *Include inst*. Now select the instances you want in the schematic. As you click the instances, the instance Name will appear in the list.
8. In the *Truncate* field, type in the number of traces you want to view. The instances with the largest noise contributions will be plotted. The gain from all the noise sources to the output is plotted. Only the noise sources from Q0 are visible. For a table that lists the noise parameter for each device type, please see page 123 (77 of 234) of this document.

© 1994-2020
All Rights Reserved.

Commonly Used Pnoise Options

TD Noise Algorithm Version=dts: is the default and is required for most oscillators that have long time constants, perhaps from a Low Drop-Out (LDO) regulator. Either take the default or set *TD Noise Algorithm Version* to *dts* for every circuit.

© 1994-2020 All Rights Reserved.

Lorentzian

There are two ways of looking at phase noise. One way is to think of measuring the spectral output of the oscillator with a network analyzer. In this case, as the frequency gets close to the oscillator frequency, the amplitude starts going up, levels off, and then drops as the oscillator frequency is passed. In this case, the noise cannot be larger than the oscillations themselves, which is 0 dBc. If this is how you view phase noise, set the *lorentzian* option to *yes*.

If you think in terms of jitter, in one cycle of the oscillator output, because of the noise, you can calculate one standard deviation of the timing. If you go two cycles, and you integrate the noise with respect to time, you get twice as much noise power, or about 1.4 times the noise voltage or jitter. When you get to an infinite number of oscillator cycles, the jitter also becomes infinite. As you increase the time, you are lowering the noise frequency, and in this case, as the time becomes large, the frequency becomes very low. At infinite time, you get infinite jitter, and this occurs at zero offset from the carrier. Jitter is just another way of looking at phase noise. In other words, the phase noise can easily exceed the carrier amplitude (0 dBc) as the offset frequency becomes small because it is heading to infinity at zero frequency offset. If this is how you view phase noise, set the *lorentzian* option to *no*.

lorentzian=yes means calculate the leveling off. **lorentzian=cornerfreqonly** calculates the continuously rising phase noise, but it places a marker on the phase noise curve at the frequency where the phase noise would level off. *no* causes the phase noise to continue to rise as the offset frequency approaches zero.

Comparison Between Lorentzian Yes and No

As noted above, lorentzian causes leveling off of the phase noise at low frequency.

Note that when *lorentzian=no*, the phase noise curve goes above 0 dBc. This would mean that the noise is larger than the amplitude of the oscillator, which is not physical. This is because pnoise is a small-signal analysis that does not recognize the large-signal limits of

© 1994-2020 All Rights Reserved.

the circuit like clipping. It is similar to AC where you can specify an input magnitude of 1 megavolt, and see 10 megavolts on the output.

or more information regarding the other pnoise options, refer to the Commonly Used Pnoise Options on page 953.

© 1994-2020
All Rights Reserved.

To plot cycle jitter or cycle-to-cycle jitter

1. On the *Direct Plot Form*, select *pnoise* jitter from the *Analysis* section.

2. Select *Jc* or *Jcc* from the *Function* section.

   The frequency range will default to the range set in the *Choose Analyses* form.

3. Click *Plot*.

© 1994-2020
All Rights Reserved.

The jc or jcc calculation is shown as a label that is added to the waveform tool.

Noise Type=sampled(jitter)

This analysis gives an instantaneous measurement of jitter at a threshold voltage on the timedomain waveform. This is useful for measuring the jitter of an oscillator that is driving a digital component.

© 1994-2020 All Rights Reserved.

1. First, set up the pss analysis. For more information on this, see Overview of Periodic Steady-State (pss) Analysis on page 647.

© 1994-2020 All Rights Reserved.

2. Next, set up the pnoise *Choosing Analyses* form, as shown below.

© 1994-2020
All Rights Reserved.

On the pnoise *Choosing Analyses* form:

- Select *sampled(jitter)* from the *Noise Type* drop-down list.
- Select *Edge Crossing*.
- Specify a threshold voltage for the time-domain waveform in the *Threshold Value* field.
- Specify a crossing direction.
- The stop frequency should be very close to half the oscillator frequency.
- The start frequency should be approximately the frequency where the phase noise reaches 0 dBc.

Note: Edge Delay and Sampled Phase modes have very limited value for oscillators. As a result, these modes are not documented in this section.

3. Run the simulation.

After the simulation completes, select *Results* - *Direct Plot* - *Main Form* in the ADE Explorer window.

On the Direct Plot Form:

1. Select *pnoise sampled* from the Analysis section.
2. Select *Edge Phase Noise* from the Function section.
3. If desired, select *Add To Outputs*.
4. Click *Plot*.

© 1994-2020 All Rights Reserved.

© 1994-2020
All Rights Reserved.

The edge phase noise is plotted in the waveform tool, as shown below.

This plot will agree with timeaverage mode single sideband results except for small differences at large offset frequencies.

Now plot the cycle jitter (Jc) or the cycle-to-cycle jitter.

1. Select *Jc* or *Jcc* from the Function section.
2. Set the frequency range to agree with the pnoise setup.

© 1994-2020 All Rights Reserved.

1. Click *Add To Outputs*.

© 1994-2020
All Rights Reserved.

This will enter an expression into ADE that will evaluate with the cycle or cycle-to-cycle jitter in the ADE outputs section.

© 1994-2020 All Rights Reserved.

Periodic Transfer Function Analysis (PXF)

Small-signal Versus Large-Signal Analysis for Conversion Gain

One way to measure conversion gain is to apply the LO and the RF signals in harmonic balance (hb) or quasi-periodic steady-state (qpss) analysis and calculate the full large-signal response at each RF input frequency, and then plot the output as a function of the sweep frequency. While this works, because all the large-signal effects and many harmonics are calculated, the required time for the simulation can be significant.

The alternative is to use a small-signal approach similar to the xf analysis. The xf analysis runs considerably faster than running the transient at multiple frequencies because it ignores the large-signal effects. In circuits with frequency translation, the xf analysis is unusable because the calculation is a linear calculation based on the DC bias point. In pxf analysis, instead of using the DC bias point, the pss solution with just the LO is used as the basis of the calculation. The large-signal analysis calculates the harmonics of the single input that causes the frequency translation to occur, (usually the LO signal or sample clock) and then pxf is run after the pss analysis to calculate the conversion gain. The conversion gain is calculated based on the nonlinearity in the pss solution. The overall time savings can be very significant.

In pac, we apply the LO signal (or the sample clock) in the pss analysis, and then drive the input with the small-signal pac analysis and measure the mixing products that come out. While this is very intuitive, the output display can be very confusing to read when the conversion gain near several pss harmonics is desired. Pxf changes the focus to the output. One output frequency at a time is considered. Pxf is a two step process. First, the input frequencies that cause the output frequency are calculated. For example, imagine a system with a 1GHz LO signal, and a 1MHz output signal. In PXF, we specify 1MHz as the output frequency. In this system, inputs at 1MHz, 999MHz, 1001MHz, 1999MHz, 2001MHz, and so on will cause 1MHz to be produced at the output. Next, the forward conversion gain from all those input frequencies is calculated. Since the input frequencies are different, the conversion gain is different for every input frequency. Using pxf, you can measure the conversion gain near a number of LO harmonics (or clock harmonics) at the same time.

© 1994-2020 All Rights Reserved.

Example: Conversion Gain (Down conversion)

Consider the double-balanced diode mixer shown below.

The RF input source has the frequency and amplitude set to variable names in order to allow changing the test setup without needing to change the circuit. This is suggested for all the sources used in the circuit that might need to be changed. This allows changes in frequency or amplitude from the ADE Explorer environment without the need to change the schematic itself.

© 1994-2020 All Rights Reserved.

The property list is shown below.

The easiest way to get the variables list into ADE Explorer is to select *Variables - Copy From Cellview*. To set the values, select the value field to the right of the variable, type in the desired value, and press *Enter*. When you have set the variables, if you want to add the

© 1994-2020 All Rights Reserved.

values back into the cellview, in the ADE Explorer window, select *Variables - Copy To Cellview*, and then perform a *Check and Save* in the schematic.

In this example, the LO frequency is 1GHz and its amplitude is 1.4 volts peak. This is +13 dBm, which is a common drive level for diode mixers. The design RF input frequency is 1.1GHz, but it is disabled because the amplitude is set to 0 (zero). Setting either the frequency or the amplitude to zero disables the production of that waveform from the port.

The basic strategy is to apply the signal that causes the frequency translation in the pss analysis, and then measure the forward conversion gain using pxf analysis.

© 1994-2020 All Rights Reserved.

The pss analysis needs to be set up first. The example below shows a single input at 1GHz with 10 harmonics. For more information on setting up the pss analysis, see the pss section at the beginning of this chapter.

© 1994-2020
All Rights Reserved.

Next, the pxf analysis needs to be set up. Like xf, the frequency range specified in the *Choosing Analyses* form is the **output** frequency range. *Maximum sideband* in this example is set to the number of harmonics in the pss analysis so that all the conversion gain terms from all the pss harmonics are present in the output of pxf.

© 1994-2020
All Rights Reserved.

Once the analyses are set up, the simulation can be run. When the simulation finishes, select *Results - Direct Plot - Main Form* in the ADE Explorer window.

The example below shows a typical *Direct Plot Form*.

The pxf *Direct Plot Form* is a bit non-intuitive. The *Choosing Analyses* form specifies the output frequency range and the output node. Pxf calculates the transfer functions from all the independent sources in the circuit to that output node. Therefore, to plot, you select the source in the circuit that you want the forward conversion transfer functions to be displayed. When the source in the schematic is selected, the waveform is displayed.

© 1994-2020 All Rights Reserved.

Note that all the short line segments are the forward conversion gain from the selected source (the input source in this case) to the output. Because the sweep was a log sweep, the data is displayed on a log plot.

© 1994-2020
All Rights Reserved.

To make the plot more useful, set the X-Axis to linear. Move the mouse cursor over one of the numbers on the X Axis, click the right mouse button, and deselect *Log Scale*.

The LO frequency is 1GHz. To measure the conversion gain, go to the **input** frequency on the X Axis, and read the gain from the display. Note that the gain above and below 1GHz is just less than -5dB. The rectangle in red above shows where to look. The gain near 3GHz is just above -20dB. The green rectangle above show where to look. The gain near 5GHz is about negative 35dB. The gain near the even harmonics is quite small.

Now position the markers at the desired input frequency for the conversion gain measurement. To do this, move the mouse cursor near the trace at the desired input frequency, and type `m`.

© 1994-2020
All Rights Reserved.

As shown in the figure below, the conversion gain is about -6dB above and below the 1GHz LO frequency. This is very common for diode mixers.

© 1994-2020
All Rights Reserved.

Example: Conversion Gain (Up Conversion)

The process is similar for an up-conversion mixer. In this case, the output frequency is just above the first harmonic of the pss analysis, as shown below.

© 1994-2020
All Rights Reserved.

The conversion gain plot is shown below. Note that the gain is very similar from near 0 Hz and near the image at just above 2GHz.

© 1994-2020
All Rights Reserved.

Overview of Simulation Capabilities

The primary application of pxf is to measure conversion gain. Pxf is a small-signal analysis. It calculates the small-signal conversion gain. If you need a large-signal measurement of conversion gain, then use a large-signal analysis like harmonic balance (hb) or quasi-periodic steady-state (qpss) analysis. Pxf measures gain in a way that requires some thought for most engineers who are accustomed to connecting an input, driving the system, and measuring what comes out. This is the idea of pac, where we capture the large-signal operating point in pss, and then drive a small-signal input, and measure the mixing products that are produced. If you think of this methodology, to measure the conversion gain around several harmonics usually requires a separate simulation near each harmonic.

Pxf was created to allow the measurement of the conversion gain transfer functions near all the pss harmonics in the same simulation run. In pac, we have one input frequency at a time, and output mixing products at multiple output frequencies are calculated. In pxf, we focus on one output mixing product at a time, and measure the input frequencies that cause that output frequency, and measure the forward conversion gain from all those input frequencies to the single output frequency. Multiple output frequencies are usually analyzed in a pxf analysis. Unlike pac where the output can be seen at every node in the circuit, pxf calculates the gain only from all the independent sources in the circuit. Therefore, to measure a conversion gain, you select the source instance from the *Direct Plot Form* instead of selecting a net.

For example, imagine a mixer with an LO frequency of 1GHz. Now, imagine the down conversion case where the output is 1MHz. Given the LO frequency, inputs at 1MHz, 999MHz, 1.001GHz, 1.999GHz, 2.001GHz, 2.999GHz, 3.01GHz, and so on cause an output at 1MHz. Depending on the circuit, all those different input to output paths have different gain from each input frequency to the output at 1MHz. In pxf, you specify the highest pss harmonic to calculate the transfer functions from in the maximum sideband field in the *Choosing Analyses* form. All these different input frequencies are called sidebands in pxf.

Pxf has three different modes, which are set by the selection in the *Specialized Analysis* section of the *Choosing Analyses* form. *Specialized Analysis = None* measures the conversion gain that occurs given the nonlinearity created by large-signal in the circuit. *Timeaverage* allows the measurement of AM to PM conversion for circuits, and is useful for measuring the power supply ripple to output phase transfer function. *Sampled* is seldom used, and supplies a measurement of the instantaneous gain when the gain is repeatedly measured at a single phase (point in time) of the periodic large signal.

PXF used with Harmonic Balance PSS

When harmonic balance is used for the pss analysis, the harmonics of the circuit are calculated directly by the harmonic balance engine.

© 1994-2020 All Rights Reserved.

Think of a perfectly linear circuit with a single output frequency to be analyzed. Because the circuit is perfectly linear, in the pss analysis when harmonic balance is used, the frequency domain solution only has the harmonic at the input frequency at every net in the circuit. No other harmonics are created because the system is perfectly linear. Because the circuit is linear, a gain function exists for the case where there is no frequency translation. Because the circuit is perfectly linear, no frequency translation can occur. Therefore, all the conversion gain terms have zero gain.

Now imagine a real circuit with a single small input amplitude signal at 1GHz. In the frequency domain with harmonic balance selected in pss, 1GHz is present along with 2GHz, 3GHz and the other higher order harmonics. Because the input signal is small, the signals at the harmonics of 1GHz in pss are very small. If 100MHz is set as the output frequency, the conversion gain is small from 900MHz and 1.1GHz. Since the higher harmonics are even smaller than the amplitude of 1GHz, the conversion gain form 1.9 and 2.1HGHz along with the higher harmonics plus and minus 100MHz are also very small.

The conversion gain might look like the figure below. Only the first three harmonics are shown. The black harmonics are the pss harmonics, and the red mixing products are pxf conversion gain measurements, which are called sidebands.

© 1994-2020
All Rights Reserved.

Now imagine that the 1GHz signal is larger. If the input goes up 10dB, the first harmonic goes up 10dB, In the frequency domain, if harmonic balance is selected in pss, the second harmonic would come up 20dB, the third harmonic would come up 30dB, and so on. If the conversion gain is calculated to 100MHz, the conversion gain is larger because all the harmonics have become larger. The circuit is more nonlinear. This is illustrated below. The black harmonics are the pss harmonics, and the red conversion gain calculations are pxf input frequencies with conversion gain measurements, which are called sidebands.

Thus, the harmonic content from pss harmonic balance is a measure of the nonlinearity of the circuit, and forms the basis of the calculation for the conversion gain in pxf.

PXF used with Shooting PSS

Shooting is a variation of the transient analysis where the individual timepoints are calculated, and the entire waveform is iterated to calculate the steady-state waveform. The transient algorithm works by using all the equations at all the nodes in the circuit where the assumption is made that the sum of the currents is zero at every node and at every timepoint. Ohm's law is observed, and the equations look like `I=V*G` where `G` is the conductance. In general, the matrix form of the equations is constructed, so the system of simultaneous equations

© 1994-2020 All Rights Reserved.

becomes `[G]*[V]=[I]` where `[G]` is a square matrix, and `[V]` and `[I]` are column matrices. To solve for [`V`], the `[G]` matrix is inverted and this matrix multiplies the `[I]` matrix to solve for all the voltages on all the nodes. At the end of the pss analysis, all the solution matrices are saved as data to be passed to the small-signal analyses. The result is that the conductance is calculated and saved for every nonlinear component in the circuit at every timepoint.

For the following circuit:

© 1994-2020 All Rights Reserved.

Below is the I-versus-V curve with three different values for the input voltage. The center, positive peak of the input, and the negative peak of the input are shown. The input amplitude is small.

In the above figure, the amplitude of the 1 GHz signal causes small variations in the slope of the tangent of the nonlinear curve as the input signal is varied in amplitude. The slope is the conductance. Because the circuit has a nonlinear device, the conductance of the device varies slightly from timepoint to timepoint in the pss analysis. Because the input signal at 1.0GHz is small, the variation in conductance from timepoint to timepoint is small. The slope changes just a little bit as the input signal is moved over its range.

Now imagine that the 1GHz signal is 10dB larger. A 10dB change is a linear factor of about 3.16. The larger amplitude of the input causes a larger portion of the nonlinear device curves

© 1994-2020 All Rights Reserved.

to be traversed, and thus the variation in conductance becomes larger. The circuit becomes more nonlinear, and the conversion gain is larger. This is illustrated below.

Therefore, the variation in the conductance of the devices from timepoint to timepoint in PSS Shooting contains information about how strongly a circuit translates frequencies, and this is used by the pxf small-signal analysis to calculate the small-signal conversion gain.

© 1994-2020 All Rights Reserved.

Normal Conversion Gain Measurement (Specialized Analysis = None)

This mode is used to make a normal conversion gain measurement.

The figure above illustrates a down-conversion case where the output is just above zero Hz.

In the figure above, the top spectrum in black represents the harmonics created by the pss large-signal analysis. The negative and positive frequencies are shown and the DC component is in the center. The colored arrows show the transfer functions from the different input frequencies to the output frequency that is shown on the bottom axis. Pxf takes the maximum sideband property from the pxf *Choosing Analyses* form, and calculates the specified sidebands, which are input to output conversion gain transfer functions. In this example, maximum sideband is set to three.

The sideband numbers indicate which harmonic number is being mixed with for the transfer function calculation. Zero means that there is no frequency translation. One is the mixing product that occurs with an input frequency shift of +1 times the pss beat frequency. Mixing frequencies to the left have negative numbers because the input frequency that causes the output is more negative than the output frequency.

For example, imagine a large-signal input at 1GHz in the pss analysis and an output frequency of 100MHz to be calculated. With no mixing, there is a gain from the sources in the

© 1994-2020 All Rights Reserved.

circuit to the output of the circuit at 100MHz. This is called the zero sideband because there is zero frequency shift from the input to the output. This is not the same measurement as a measurement in the ac analysis because the ac only includes the DC operating point, and the pxf analysis takes into account the circuit with the LO or clock signal applied.

The input frequencies that mix with the first harmonic at 1GHz to provide an output at 100MHz are at -900MHz and 1.1GHz. The frequency that is below the zero sideband is the -1 sideband and the frequency above the zero sideband is the +1 sideband.

Similarly, there are terms that mix with the second and third harmonics.

Usually, we think of positive frequencies only. This is shown above. The negative frequency inputs are reflected up into the positive frequencies. This is the default for pxf.

Your application sets the number of sidebands that you need. If you have specs through the fifth harmonic, you need to set maximum sidebands to 5.

© 1994-2020 All Rights Reserved.

Frequency Sweep

This analysis is like xf, where the **output** frequency is varied. The output frequency range to be analyzed is specified in the pxf *Choosing Analyses* form. Pxf then calculates the conversion gain terms.

In addition to the frequency, there is a selection called *Sweep Type*. The choices are *relative* and *absolute*. When *absolute* is selected, the frequency range is used directly with no frequency conversion. When *relative* is selected, a *Relative Harmonic* field appears. The frequency sweep is shifted by the specified frequency of the harmonic in the pss analysis. For example, assume that the pss is at 1GHz, and a log sweep is desired above the first harmonic of the pss analysis. In this case, you need to select *relative* sweep, and specify 1 in the *Relative Harmonic* field. Next, type in 1K to 100M for the frequency range with a log sweep and 3 to 5 points per decade.

The sweep will start at 1G + 1K, and sweep to 1G + 100M using log spacing from 1K to 100M. This is illustrated below.

© 1994-2020
All Rights Reserved.

To use a log sweep below the first harmonic, use the same frequency range, and type in -1 as the relative harmonic. This is illustrated below.

Maximum Sideband

Sideband is the name of the pxf conversion gain terms. The sideband number is the pss harmonic number that is mixed with. If the sideband number is negative, the input frequency is more negative than the output frequency. If the sideband is positive, the input is more positive than the output frequency.

Conceptually, an infinite number of conversion gain terms exist when a single output frequency is analyzed in a nonlinear system. From the practical point of view, usually only a small number of conversion gain terms need to be measured near the LO frequency and a small number of harmonics.

There is a property in the *Choosing Analyses* form called *Maximum sideband* that is used to define how many mixing products should be calculated. If *Maximum sideband* is 0, the pxf analysis will only contain the frequency that appears without mixing or aliasing of any type. This is not equivalent to a linear AC analysis because pxf takes into account the instantaneously varying nature of the LO or sample clock signal because of the large signal being present in the pss analysis.

© 1994-2020 All Rights Reserved.

When *Maximum sideband* is 1, the inputs without frequency translation and the inputs that mix or alias with the first harmonic of the pss analysis are present in the pxf analysis. When *Maximum sideband* is 10, the pxf analysis includes mixing through the 10th harmonic of the pss analysis. When shooting is used for the pss analysis, the number of sidebands can be any number up to four times the number of harmonics specified in the pss analysis. When harmonic balance is selected, up to the number of harmonics in the pss analysis can be specified.

Setting Harmonics and Sidebands

Shooting

The pss analysis, when shooting is selected, has a minimum of 200 timepoints, so it inherently has frequency domain content through the 100th harmonic of the pss beat frequency. Maximum harmonic is a pss post-processing property that specifies how many harmonics to calculate in the Fourier transform of the waveform in pss. If the number of harmonics is zero, it just means that no frequency domain information should be calculated in the simulation. If the number of harmonics is 10 or less, there is no effect on the PSS waveform. When maximum harmonics gets larger than 10, the minimum number of timepoints in the pss analysis is increased. 20 timepoints in the period of the highest harmonic are forced. Thus, if maximum harmonics is raised above ten, the pss waveform gets more timepoints which increases the accuracy of the waveform, at the cost of longer runtimes and more memory required to run the pss simulation.

In order for the pxf analysis to correctly measure conversion gain, the wave shape at the highest mixing harmonic needs to be accurate, so with the default settings in shooting pss, up to 40 sidebands can be accurately calculated in the pxf analysis. In most cases, only a small number of sidebands needs to be calculated, and so with shooting, you can usually count on the pxf result to be accurate.

Although this is very unusual, if you need more than 40 sidebands when you are using shooting pss, you need to set the *maxacfreq* option in the pss analysis to the pss beat frequency times the number of sidebands requested in the pxf analysis, or alternatively, set the number of pss harmonics to the number of sidebands in pxf divided by four. Either method works to set the maximum timestep in the pss analysis as required for the pxf analysis to be accurate.

Harmonic Balance

When *Harmonic Balance* is selected in the pss analysis, only the harmonics specified in the pss *Choosing Analyses* form exist in the solution. In the pxf analysis, because only the harmonics specified in the pss-hb *Choosing Analyses* form exist, the maximum sideband

© 1994-2020 All Rights Reserved.

term can only be the number of harmonics specified in the pss analysis or less. Usually, when *Harmonic Balance* is selected in the pss analysis, the *Maximum sideband* field in pxf usually should be left blank which calculates all the conversion gain calculations from all the harmonics in the pss analysis. The *Direct Plot Form* is then used to display just the conversion gain terms you want.

Note that with harmonic balance, the number of harmonics (and oversample factor if you have non-sinusoidal waveforms) in the pss analysis needs to be determined. Start with a number of harmonics in the pss based on an estimate of how many harmonics the circuit actually produces with your input waveform, and run the pss and pxf analyses. Remember that for non-sinusoidal waveforms, oversample factor also needs to be set greater than one. For more information, see the pss section at the beginning of this chapter or the hb analysis at the beginning of Chapter 3. Run the analysis and make the desired measurement. Now increase the number of harmonics by roughly 50% and run the simulation again. If the answer does not change, then you had enough harmonics originally, and you might actually be able to decrease the number of harmonics from the original setting. If you have non-sinusoidal waveforms, increase the oversample factor to between four and eight before increasing the number of harmonics to find the minimum value for both oversample factor and harmonics.

Pxf is usually preferred to pac when you need to measure both the passband and the image. Below is a plot of the pxf and pac output for an image reject mixer with a 1.9GHz LO and a 100MHz IF. The passband is at 1.8GHz and the image is at 2GHz.

© 1994-2020
All Rights Reserved.

On the left is a pxf plot when the input port is probed, thus measuring the gain from the input to the output of the image reject mixer. To read the gain, go to the **input** frequency, and read the gain on the vertical scale. At the passband frequency of 1.8GHz the gain is about 8dB. In the image at 2GHz, the gain is about -70 dB. The LO frequency is at the junction of the red and blue traces.

To measure passband and image gain in pac, the input frequency starts at 1.7GHz and stops at 2.1GHz. At 1.7GHz, the output frequency is -200MHz, but pxf reflects the negative frequencies to the positive frequency domain by default. The trace starts at point A above, and sweeps down to zero at point B. The input frequency is 1.9GHz and the output is zero Hz. As the input continues above 1.9GHz, the output frequency becomes positive, and follows the curve to point C. Most people find the curve from pac to be more confusing than the curve from pxf, so if you need to measure both the passband and the image, or if you need to measure the conversion gain near the pss harmonics, pxf is recommended.

Note that in the pac result above, you cannot just add the gain values at a specific frequency. If the frequency was 100MHz, you do have two different gain values of about +5dB and -70 dB. The output frequency for the +5dB curve is -100M, and the output for the -70dB curve is +100MHz. Since the frequencies are different, you cannot just add the two gain numbers.

Pxf can measure the gain near may harmonics at the same time by increasing the maximum sideband parameter in the *Choosing Analyses* form. Below is an example of conversion

© 1994-2020 All Rights Reserved.

gain through the fifth harmonic. Your application will determine the number of sidebands to set.

Note that the conversion gain is still relatively high near the third harmonic.

Pxf can also be used to measure the conversion gain from the battery that supplies power to the output. All the gain and conversion gain terms should be below unity, or a potential

© 1994-2020
All Rights Reserved.

oscillator is created. In the plot below, the gain from the bias source to the output at the IF frequency of 100MHz is above 0dB. This circuit is a potential oscillator.

© 1994-2020
All Rights Reserved.

Pxf is useful for up-conversion applications as well, with the limitation that a small-signal gain measurement is made. This is shown below.

The difference is that the output frequency is just above the first harmonic. The rules for the sidebands are the same. -1 means that the input frequency is shifted down from the output frequency by -1 * pss beat frequency.

© 1994-2020
All Rights Reserved.

Usually, we think in terms of the positive frequency domain, which is shown below.

© 1994-2020
All Rights Reserved.

AM to PM measurement (Specialized Analysis = Modulated)

This is used to measure the am to pm conversion. Normally, it is used for oscillators to measure the am to pm conversion from the power supply, but it can also be used for driven circuits. In both cases, remember that pxf is a small-signal analysis. If you need a large-signal measurement, use the transient, hb, or pss analysis.

In modulated analysis, the output type should be `ssb/am/pm` so that all the gain terms can be calculated. Generally, the AM to PM gain is desired.

Conceptually, if the **output** is ssb (single sideband) then the PM output cannot be measured.

The gain from SSB, AM, or PM inputs to the output frequency is the only gain measurement available.

© 1994-2020
All Rights Reserved.

When the output is AM, both the sidebands above and below the chosen harmonic are calculated.

© 1994-2020
All Rights Reserved.

This can measure SSB, AM, or PM to an AM output.

When the output is PM, the SSB, AM, or PM to PM gain can be measured. Generally, the AM to PM gain is desired.

© 1994-2020
All Rights Reserved.

Sampled PXF

This analysis is seldom used.

The basic idea is to measure the conversion gain as if the conversion gain was repeatedly measured at the same phase (instant in time) in the periodic large signal that is driving the circuit over an infinite number of cycles of the large signal. This measurement is impossible to duplicate in the lab. The primary application is for measuring the gain from the power supply to the output which is driving a digital circuit or a comparator that is triggered at a specific voltage level.

© 1994-2020 All Rights Reserved.

ADE Implementation

This section jumps from the theoretical to the practical with examples of how to use the individual analysis types we just covered. The focus is only on the pxf *Choosing Analyses* form so you can see where the individual settings are made. For examples with all the steps shown, see the examples section that follows this section.

© 1994-2020 All Rights Reserved.

PXF for a Normal Conversion Gain Measurement

Here is an example of the *Choosing Analyses* form for a normal conversion gain measurement.

© 1994-2020
All Rights Reserved.

1. At the top of the form the PSS beat frequency of 1.9GHz is shown.
2. Sweeptype can be relative or absolute. The default for driven circuits is absolute, and for oscillators it is relative. The selection that is being used is shown to the right of the sweeptype selection.
3. The frequency sweep range is always an **output** frequency range. It is not a sweep in the traditional sense of connecting an input, sweeping over a frequency range, and measuring the output. Instead, it is an output frequency range to be analyzed. At each output frequency in the sweep, the input frequencies that will mix with the pss harmonics to cause that output frequency are calculated, and the forward conversion gain is calculated.
4. Linear and log spacings are provided. Generally, *Auto* should be avoided because it will always calculate 50 frequency points. Usually, a much smaller number is needed, so it is recommended that you set *linear* or *log*, and control the resolution yourself.
5. Although there are choices for the Sidebands selection, it is usually better to leave the setting at *Maximum sideband*, and if you need a small subset of the output results, use the *Direct Plot Form* to filter the results. For shooting, you can use up to 40 sidebands with no additional settings. If you need more than 40 sidebands, either set the number of pss harmonics to the number of sidebands divided by 4, or set the pss option *maxacfreq* to the pss beat frequency times the number of sidebands in pxf. For harmonic balance, choose a number of harmonics in the pss form, and leave the *Maximum sideband* field blank. This will calculate the conversion gain from all the harmonics in the pss analysis. For harmonic balance only, run the simulation, and make the measurement in pxf. Now increase the number of pss harmonics by about 50% and run again. If the pxf result changed significantly, then you need more harmonics in pss. If it did not change, use the smallest number of harmonics that results in an stable measurement.
6. The output can measure either a voltage or a current. The voltage measurement is the default. Specify the output node or nodes in the circuit. If the output is a current select probe for the output, add an iprobe from analogLib (a current probe) in series with the output in the schematic, and specify the instance name in the *Choosing Analyses* form.
7. When *Specialized Analysis* is set to *None*, a normal conversion gain measurement is made.

© 1994-2020 All Rights Reserved.

PXF Modulated (AM to PM Conversion)

Here is an example of a modulated *Choosing Analyses* form.

© 1994-2020
All Rights Reserved.

1. The functions of the form above the *Specialized Analyses* section are the same as for a normal conversion gain measurement.
2. In the *Specialized Analyses* section, select *Modulated*. The bottom part of the form redraws.
3. In general, set the *Output Type* to *SSB/AM/PM*. This allows all the different transfer functions to be calculated, with only a small performance penalty.
4. In modulated, the input and output frequencies need to be identified. The best way to do this is to click the *Choose* button to the right of the field, select the input frequency from the list of available frequencies, and click *OK*. Then repeat this action for the output frequency range, as shown below.

© 1994-2020 All Rights Reserved.

PXF Sampled

Sampled pxf is used to measure the conversion gain at a specific point in time of the periodic signal during the pss analysis. In general, this analysis is applied to digital circuits so that the instantaneous conversion gain from the power supply to the output can be measured. Below is the time-domain waveform of a 3.3 volt digital logic circuit.

In this case, a threshold of 1.65 volts would be appropriate.

© 1994-2020
All Rights Reserved.

Below is an example of a sampled setup for a 3.3 volt logic circuit.

© 1994-2020
All Rights Reserved.

1. The functions of the form above the *Specialized Analyses* section are the same as for a normal conversion gain measurement.
2. In the *Specialized Analyses* section, select *Sampled*. The bottom part of the form redraws. If your output is a current, add an iprobe (current probe) from analogLib in series with your output in the schematic, and select the iprobe as the probe selection. If you have a voltage, select voltage, and specify a node or nodes in the schematic.
3. The normal use of the form is to specify a voltage threshold for the measurement in the threshold field, and not specify additional timepoints. The example above is for a 3.3 volt logic circuit where the switching threshold is 1.65 Volts.
4. The *Crossing Direction* can be set to *all*, *rise*, or *fall*.
5. In the case of multiple crossings, as shown in the first pss waveform shown above, all the crossings will be calculated with a maximum of 16 by default. If you have more than 16 crossing events, set the number of crossings you have in the *Maximum Samples* field. In this example, there are two crossing events, so no entries are required in the *Maximum Samples* field.
6. If you have specific pss times you want to measure, specify a list of pss simulation times in the *Additional Timepoints* field. Separate the entries with a space. An example of using only the timepoints is shown below. A threshold of -10V is well outside the

© 1994-2020 All Rights Reserved.

anticipated voltage on the output node which disables the timepoint selection using a voltage threshold.

© 1994-2020
All Rights Reserved.

PXF Options

Below is the Pxf options form.

© 1994-2020 All Rights Reserved.

The below options are occasionally used.

Solver

The solver option is used only for driven circuits. The default solver is the *turbo* solver.

When hb is used as the engine in pss, leave this option at the default.

When shooting is used, sometimes when the pxf output frequency is very close to the frequency of one of the harmonics in the pss result, warning messages will appear in the pxf output warning that the accuracy might not be good enough. If you see these messages, select the *std* solver, which has better ability to handle frequencies that are very close to a harmonic in the pss, but which takes longer to run than the *turbo* solver.

Oscsolver

The oscsolver option is used only for oscillators. The default solver is the *turbo* solver.

When hb is used as the engine in pss, leave this option at the default.

When shooting is used, sometimes when the pxf output frequency is very close to the frequency of one of the harmonics in the pss result, warning messages appear in the pxf output warning that the accuracy might not be good enough. If you see these messages, select the *std* solver, which has better ability to handle frequencies that are very close to a harmonic in the pss, but takes longer to run than the *turbo* solver.

For very large oscillator designs, the *ira* oscsolver may take less memory and run faster with no loss in accuracy, however, it is less robust for convergence compared to *turbo* or *std*.

Hbprecond_solver

This option is only available only when harmonic balance is selected in PSS and APS is used.

The basic solver is the only solver available in standard Spectre. *autoset* is the default solver when APS is used. This solver is faster, but occasionally stagnates. When stagnation is detected, APS automatically switches to the basic solver, and prints a message in the Spectre output window. If you have stagnation, it will save a small amount of time to set this option to *basicsolver*.

© 1994-2020 All Rights Reserved.

Freqaxis

freqaxis specifies whether you want to see the negative frequency axis or not.

© 1994-2020 All Rights Reserved.

The analysis calculates the input frequencies and forward conversion gain, so *in* and *absin* are reasonable choices. *in* displays the negative frequency axis. *absin* (absolute value of the input) displays positive input frequencies. This is the default.

© 1994-2020
All Rights Reserved.

The input frequencies are plotted on the positive frequency axis.

To read the conversion gain, go to the **input** frequency on the curve, and measure the conversion gain.

Below is an example of *freqaxis* set to *in*.

© 1994-2020
All Rights Reserved.

Note that the data is the same as before. It is just displayed with the negative frequency axis present. The negative frequencies have not been reflected up to the positive frequency domain. The waveform display shows the negative frequency axis now.

If you select *out*, all the inputs at the different frequencies are plotted on the same output frequency range scale. This is not recommended.

© 1994-2020
All Rights Reserved.

The waveform tool shows all the different input frequencies on the same output frequency scale. Again, this is not recommended.

The following options are not normally used.

Tolerance

Leave this option at the default value.

Pxf uses an iterative solver to calculate the conversion gain terms. Any iterative solver needs an error tolerance to specify when to stop iterating because the solution is accurate enough. The tolerance option specifies that accuracy for pxf. For shooting, the default tolerance is 1e-9. For driven circuits where HB is the pss engine, the default is 1e-6. For oscillators where hb is selected for the pss engine, the default is 1e-4.

Gear_order

Do not change this option.

© 1994-2020 All Rights Reserved.

Resgmrescycle

Leave this option at the default.

Annotate

This option controls the level of detail in the Spectre output log. Less detail is on the left. More detail is on the right.

Stimuli

This option is not implemented at the current time.

Save

This option is not implemented at the current time.

Nestlvl (Shooting and Harmonic Balance)

This option is not implemented at the current time.

AdditionalParams

In some cases when shooting pss is used for an oscillator, PXF results that are close to a pss harmonic may be questionable. In this case, type osc_version=dts in the additionalParams field to solve the issue.

additionalParams is typically used for new features that are being beta tested. This field expects `keyword=value` pairs.

For more information about the other options, type `spectre -h pxf` at the command prompt in a Unix shell window.

© 1994-2020 All Rights Reserved.

Examples for Driven Circuits

PXF Normal Conversion Gain Measurement

Consider the following circuit:

The circuit is a double-balanced down-conversion mixer with a 1.9GHz LO.

© 1994-2020
All Rights Reserved.

The *Edit Object Properties* form for the input port is shown below.

© 1994-2020
All Rights Reserved.

Using variables to set the frequencies and amplitudes is suggested because you can change the circuit setup by changing variable values in ADE Explorer without needing to change the circuit itself.

ADE Explorer has the variables set to appropriate values, as shown below.

This setup causes the LO to be driven, and both RF inputs to be turned off for the pss analysis by setting the RF frequencies to zero. You can disable a signal for the pss analysis by setting its frequency to zero.

Using PXF Analysis for Conversion Gain Measurement

Set up the pss analysis with just the large signal that causes the frequency translation to occur, which is generally the LO signal or the sample clock. For more information, refer to *Overview of Periodic Steady-State (pss) Analysis* section at the beginning of this chapter. Regarding whether to use shooting or harmonic balance, pick the analysis that

© 1994-2020 All Rights Reserved.

produces the smallest runtime or best convergence. In general, shooting is recommended for systems that have sharp edges, and harmonic balance is recommended for systems that are more sinusoidal.

© 1994-2020 All Rights Reserved.

© 1994-2020
All Rights Reserved.

Set up the PXF analysis, as follows.

1. Specify an appropriate **output** frequency range for your circuit.
2. Generally, either linear or log spacing should be used. Auto usually takes more frequency point than are required, and thus extends the runtime.

© 1994-2020
All Rights Reserved.

3. For the pxf analysis, as a general recommendation, set the *Maximum sideband* term to the same value you uses for the maximum number of pss harmonics. Alternatively, the *Maximum sideband* field could be set to the highest harmonic of the pss analysis that mixing products are desired to be calculated. This is generally determined by the application. If you only need one or two output mixing product, and you are using shooting in the pss analysis, choose *Select from range*, and highlight the sidebands you want to calculate. Setting *Maximum sideband* for pxf is usually not critical as long as enough sidebands are calculated to measure the desired input frequency and forward conversion gain (called a sideband). The *Direct Plot Form* allows the plotting of the individual input sidebands as will be shown later.

4. Set the output nodes, as shown in the example. If the *Negative Output Node* field is left blank as shown above, the global ground node is used by the simulator for the reference node. If you have an output current, add an iprobe from analogLib in series with the output, and click probe in the output section. Then specify the iprobe in the *Output Probe Instance* field.

5. Now run the simulation.

© 1994-2020 All Rights Reserved.

6. After the simulation completes, select *Results - Direct Plot - Main Form*. The *Direct Plot Form* is displayed.

In the *Direct Plot Form*:

7. Select *pxf* from the *Analysis* section.
8. Select *dB20* from the *Modifier* section.
9. Select the voltage or current source from the schematic. Note that when you plot pxf results, you are selecting an instance, not a node. You will get the conversion gain results for the source you select in the circuit. The example below shows the conversion gain from the input source in the circuit.

© 1994-2020 All Rights Reserved.

The waveform window is displayed, as shown below.

Since the *Sweeptype* was *log*, the data is displayed with a log axis in the waveform tool. In some cases, a linear X Axis is easier to read. To accomplish this, move the mouse cursor over one of the numbers on the X Axis and right-click. Move to Log Scale and click to deselect log. This is shown below.

© 1994-2020
All Rights Reserved.

To read the conversion gain, move to the desired **input** frequency, and read the gain off the Y axis. The gain at 1.91GHz is about 15dB. The place to look is in the red rectangle above. Alternatively, position your mouse cursor on the trace at the input frequency, and read the value in the cursor readout. You can also place a marker at the input frequency to read out the conversion gain. To do this, move your mouse cursor near the trace at the input frequency, and type m.

© 1994-2020
All Rights Reserved.

10. The conversion gain near the third and fifth harmonics of the LO is also reasonable high as well. The gain near the third harmonic is near -5dB, and the gain near the third harmonic is near -25 dB. See the red and blue boxes below.

11. In the *Direct Plot Form*, select *New Win* from the *Plotting mode* drop-down list and select the DC source that is supplying power to the circuit.

© 1994-2020
All Rights Reserved.

The waveform window is displayed, as shown below.

Note that the gain is about +5dB at 150MHz. This circuit is a potential oscillator.

© 1994-2020
All Rights Reserved.

PXF Modulated (AM to PM conversion)

Consider the digital circuit below.

1. The input source has variables to define the frequency. This allows the frequency to be changed from the ADE Explorer environment without needing to change the circuit. The properties list is shown below. The rise and fall times are set to a small fraction of the period of the signal. This is done so the rise and fall times do not produce nonphysical,

© 1994-2020 All Rights Reserved.

excessively fast risetimes, which cannot be realistically produced in a circuit anyway. The pulse width is set to produce a 50% duty cycle square wave.

© 1994-2020
All Rights Reserved.

2. The default values for the variables are set in ADE Explorer. The input frequency is 2.5GHz.

© 1994-2020 All Rights Reserved.

3. To run a pxf analysis, first a pss analysis needs to be run. For more information, see the pss section at the beginning of this chapter.

© 1994-2020
All Rights Reserved.

4. Now set up the pxf modulated analysis.

© 1994-2020
All Rights Reserved.

5. First, select *Modulated* at the bottom of the *Choosing Analyses* form. This forces relative Sweeptype.
6. The objective is to measure AM to PM conversion on the 2.5GHz output when there is power supply ripple. Remember that in pxf, the frequency range is an **output** frequency range. Accordingly, the relative harmonic is 1, and the output frequency is analyzed just above the output at 2.5GHz.
7. Usually, it is better to select linear or log spacing, and set your own sweep resolution. Automatic usually takes more points than are needed, thus extending the runtime.
8. Set the number of sidebands equal to the number of pss harmonics.
9. Specify an output node or nodes, as usual in pxf. If you have an output current, add an iprobe in series with the output, and click probe in the *Output* section. Then specify the instance name of the iprobe in the *Output Probe Instance* field.
10. Always set the *Output Type* to *SSB/AM/PM*. This allows all the SSB/AM/PM to SSB/AM/PM components to be plotted.
11. Click the *Choose* button to the right of the *Input Modulated Harmonic List* field. Select the input frequency as shown below. This example measures the power supply ripple at low frequency to PM output near the 2.5GHz carrier, so the input frequency range is 1K to 100M.

12. Click the *Choose* button to the right of the *Output Modulated Harmonic* field. Select the output frequency, as shown below. This example measures the power supply ripple

© 1994-2020 All Rights Reserved.

at low frequency to PM output near the 2.5GHz carrier, so the output frequency range is 2.5G to 2.6G.

© 1994-2020 All Rights Reserved.

13. Click *OK*, and run the simulation. When the analysis completes, select *Results - Direct Plot - Main Form* from the ADE Explorer window.

In the *Direct Plot Form*:

1. Select *pxf modulated* from the *Analysis* section.
2. At the current time, to measure AM to PM conversion when the input is near zero Hz, the *Input* option should be set to *SSB*.
3. Set the *Output* option to *PM*.

© 1994-2020 All Rights Reserved.

4. Select *Voltage Gain* from the *Function* section
5. Select *Magnitude* from the *Modifier* section.
6. Click the DC source in the circuit that supplies power to the circuit. The measurement plots.

7. Although the vertical axis is labeled mV/V, it is actually plotting milliRadians of phase change per volt of power supply change. 318 milliRadians is about 18 degrees of phase change for a 1 volt change on the power supply. [(0.318/6.28)*360] Note that pxf is a small-signal measurement, and this applies only for small variations of the power supply voltage.

© 1994-2020 All Rights Reserved.

Sampled PXF

Sampled pxf is seldom used. It is provided so that conversion gain measurements from the power supply to a digital output that is periodic, and sampled at the threshold point every cycle. The typical threshold is the VDD/2 switching point for further logic circuits.

Consider the digital circuit below.

1. The input source has variables to define the frequency. This allows the frequency to be changed from the ADE Explorer environment without needing to change the circuit. The properties list is shown below. The rise and fall times are set to a small fraction of the period of the signal. This is done so the rise and fall times do not produce nonphysical,

© 1994-2020 All Rights Reserved.

excessively fast risetimes, which cannot be realistically produced in a circuit anyway. The pulse width is set to produce a 50% duty cycle square wave.

© 1994-2020
All Rights Reserved.

2. The default values for the variables are set in ADE Explorer. The input frequency is 2.5GHz.

© 1994-2020 All Rights Reserved.

3. To run a pxf analysis, first a pss analysis needs to be run. For more information, see the pss section at the beginning of this chapter.

© 1994-2020 All Rights Reserved.

4. Now set up the pxf sampled analysis.

5. First, select *Sampled* at the bottom of the *Choosing Analyses* form.

© 1994-2020
All Rights Reserved.

6. The objective is to measure the conversion gain on the 2.5GHz output at the threshold crossing of the digital output when there is power supply ripple. Remember that in pxf, the frequency range is an **output** frequency range. Accordingly, the output frequency is analyzed just above the output at 2.5GHz.
7. Usually, it is better to select linear or log spacing, and set your own sweep resolution. Automatic usually takes more points than are needed, thus extending the runtime.
8. Set the number of sidebands equal to the number of pss harmonics.
9. Specify an output node or nodes, as usual in pxf. If you have an output current, add an iprobe in series with the output, and click probe in the *Output* section. Then specify the instance name of the iprobe in the *Output Probe Instance* field.
10. Click *Select* to the right of the *Net+* field and select the output net in the schematic.
11. Type in the threshold voltage. In 3.3 volt logic, 1.65 volts is appropriate.
12. Choose the crossing direction. Rising, falling, and all crossing events are selectable. This example shows all, which will calculate the transfer functions for both the rising and falling edges.
13. If you want specific times in the pss waveform to be measured, type those times into the *Additional Timepoints* field. Separate the times with a space. This is not shown in the example.

© 1994-2020 All Rights Reserved.

14. Click *OK* and run the simulation. When the analysis completes, select *Results - Direct Plot - Main Form* from the ADE Explorer window.

© 1994-2020
All Rights Reserved.

15. Select pss results, and plot the time-domain waveform. This is shown in the previous graphic. Plotting the waveform establishes which time corresponds to which edge.

16. For this example, the rising edge is about 77nsec, and the falling edge is about 275nsec.

© 1994-2020
All Rights Reserved.

17. Now select pxf results. If desired, select dB20. Select the crossing time from the list at the bottom. The example shows the rising edge.

© 1994-2020
All Rights Reserved.

18. Click the DC source in the circuit that supplies power to the circuit. The measurement plots.

© 1994-2020
All Rights Reserved.

19. To measure the conversion gain, go to the **input** frequency on the plot, and make the measurement. Below, the plot is zoomed in to the results from low frequency, like power supply ripple. At 2MHz, the gain is about -0.5dB.

© 1994-2020
All Rights Reserved.

Examples for Oscillators

Conversion Gain From Power Supply to the Output Frequency

Consider the circuit shown below.

On the *Choosing Analyses* form:

© 1994-2020
All Rights Reserved.

1. Set up the pss analysis, as shown below. For more information, refer to *Overview of Periodic Steady-State (pss) Analysis* at the beginning of this chapter.

© 1994-2020 All Rights Reserved.

2. Set up the pxf analysis, as shown below.

3. R*elative* from the *Sweeptype* drop-down list is the default when an oscillator simulation is set in the pss analysis.

4. Specify the output harmonic in the *Relative Harmonic* field.

© 1994-2020
All Rights Reserved.

5. Set the frequency range to a frequency range that is relevant for your needs. In this example, the frequency range is from 100Hz to 100MHz above the first harmonic of the oscillator frequency.
6. Select *logarithmic* from the *Sweep Type* drop-down list.
7. Specify 3 to 5 in the *Points Per Decade* field.
8. Set the *Maximum sideband* option to the same value as you have set for the maximum harmonics in the pss analysis.
9. Click *OK*.
10. Run the simulation.
11. When the simulation completes, select *Results - Direct Plot - Main Form*. The *Direct Plot Form* is displayed.

© 1994-2020 All Rights Reserved.

In the *Direct Plot Form*:

12. Select *pxf* in the *Analysis* section.

13. Select *sideband* from the *Sweep section*.

14. Select dB20.

15. Select your **input** frequency range in the list in the *Input Sideband* section.

16. Select the power supply source in the Schematic window.

The waveform is plotted, as shown below.

To read the gain, go to the input frequency, and read the gain from the Y axis. The gain is about 47dB at 100KHz in this example.

© 1994-2020
All Rights Reserved.

Modulated PXF; Measuring AM to PM Conversion and Controlling Spurious Response

The focus of this section is controlling the spurious response caused by the power supply ripple. Consider an oscillator shown below.

The first part of this section uses hb, which is equivalent to pss. The reason hb is used is because hb has the ability to run an oscillator with ripple so the spurious output of the oscillator can be measured. Pss does not have this capability. Using hb to solve for the spurious output usually takes longer than a pss with a modulated pxf analysis. Pss-pxf modulated will be used to measure the conversion gain from the power supply ripple to the output phase of the oscillator. Once the spurious output is measured with hb, then running pss and modulated pxf can be used to reduce the conversion gain from the power supply to the output, thus reducing the spurious output. Using this combination of analyses saves time.

© 1994-2020 All Rights Reserved.

1. On the left are two sources. The bottom source is the DC source that supplies power to the circuit. This source is in the yellow rectangle above. The upper source is a source that injects ripple into the simulation. This source is in the red rectangle above.
2. The ripple frequency and amplitude are set using variable names so the values can be changed in the ADE Explorer environment without needing to change the circuit. The ripple source properties are shown below.

© 1994-2020
All Rights Reserved.

3. ADE Explorer sets the frequency to 1MHz and the amplitude to 30mV peak.

© 1994-2020
All Rights Reserved.

4. First, measure what is happening with the ripple applied in the circuit. See the hb section for more details. The hb setup is as below.

© 1994-2020 All Rights Reserved.

5. Run the simulation, and plot the output spectrum.

6. Place a marker by moving your mouse cursor over the tip of the oscillator frequency, and typing m. Now move your mouse cursor over one of the spurs, and type d. This places another marker, and causes the delta readout shown above to be displayed.

   The spurious output is about 33.3dB down from the oscillator output frequency.

7. Now that we know what we really have, and how much this is above the specification, we can switch to a faster method to reduce the spurious output. This will be done using pss and modulated pxf.

8. Now measure the AM to PM conversion using pss and pxf modulated. This will tell you the gain from the power supply to the PM output.

© 1994-2020 All Rights Reserved.

9. First, disable the ripple source by setting its frequency to 0 in ADE Explorer, and disable the hb analysis used to measure the spurious response.

© 1994-2020 All Rights Reserved.

10. Set up a pss analysis with just the oscillator signal. For more information, see the pss section at the beginning of this chapter.

© 1994-2020 All Rights Reserved.

11. Now set up a pxf modulated analysis, as follows:

© 1994-2020
All Rights Reserved.

12. At the bottom of the form, set *Specialized Analysis* to *Modulated*, and set the *Output Type* to *SSB/AM/PM*.

13. At the top of the form, *Modulated* forces a relative frequency sweep. The frequency sweep range is always an output frequency range in pxf. Since the gain from the power supply ripple to the output above the oscillator frequency is desired, the output harmonic is 1 and the frequency range is just above the first harmonic.

14. It is usually better to select either a linear or a log sweep, and define your own resolution because automatic usually takes more points and therefore a longer runtime than is usually needed.

15. Set *Maximum sideband* to the same value that you used for harmonics in the pss analysis.

16. Specify the output node or current as usual. If you have an output current, add an iprobe in series with the output, and click probe in the Output section. Then specify the instance name of the iprobe in the Output Probe Instance field.

17. Now, set the input frequency by clicking *Choose* to the right of the *Input Modulated Harmonic List* field.

    The Choose Harmonic window is displayed.

18. In the Choose Harmonic window, select the frequency range and click *OK*.

19. To set the output frequency, click *Choose* to the right of the *Output Modulated Harmonic* field.

© 1994-2020 All Rights Reserved.

The *Choose Harmonic* window is displayed.

20. In the *Choose Harmonic* window, select the output frequency range and click *OK*.

21. Click *OK* in the *Choosing Analyses* form and run the simulation.

© 1994-2020 All Rights Reserved.

22. When the simulation completes, select *Results - Direct Plot - Main Form* in the ADE Explorer window.

© 1994-2020
All Rights Reserved.

23. Currently, to measure the AM to PM conversion, you set *Input* to *SSB*, and *Output* to *PM*. Select *dB20,* if desired. Select the ripple source in the schematic. The AM to PM gain is plotted.

24. Position a marker at the ripple frequency to measure the gain. The gain in this example is about 3.9dB. As you implement circuitry to reduce the ripple, you can run the pss-pxf modulated analysis in less time than it takes to run the hb analysis with both the oscillator and the ripple. When you have a ripple reduction circuit in place, you can re-run the two-tone hb analysis to verify the fix works.

© 1994-2020 All Rights Reserved.

Sampled PXF: Measuring Conversion Gain at a Threshold Crossing in a Ring Oscillator

Sampled pxf can be used to calculate the gain from the power supply to the output when the output is defined as the periodic point where the gain is measured at the same phase (time) of the oscillator as if it were sampled repeatedly at exactly the same phase (time) over an infinite number of cycles. This is the conversion gain from the power supply to a single point in the output waveform, as opposed to the averaged measurements from before. Usually, this is applied to ring oscillators where relatively sharp edges are present in the output, and the output is driving some kind of digital logic. In this case, only the point where the threshold for the digital logic on the output matters. All the rest of the points in the waveform are either above or below the threshold, and thus do not influence the threshold event time.

1. Consider the 4-stage differential voltage-controlled ring oscillator below.

© 1994-2020
All Rights Reserved.

2. First, set initial conditions on one of the stages of the ring to force a DC operating point. In the ADE Explorer window, select *Simulation - Convergence Aids - Initial condition*.

© 1994-2020
All Rights Reserved.

3. The *Select Initial Condition Set* window appears. Since this is a differential oscillator, there are two outputs from one stage. One of those outputs is set low, and the other is set high. This forces a starting condition for the tstab interval.

© 1994-2020
All Rights Reserved.

4. When the *Select Initial Condition Set* window is displayed, the nodes with the IC defined are highlighted in the schematic, as shown below. The fourth stage output is set.

© 1994-2020
All Rights Reserved.

5. Now set up the pss analysis. For more information, please see the pss section at the beginning of this chapter.

© 1994-2020
All Rights Reserved.

6. Set up the sampled pxf analysis, as shown below.

© 1994-2020
All Rights Reserved.

7. Select *Sampled* from the *Specialized Analyses* drop-down list.
8. Because the pss is set for an oscillator, relative is set for pxf. The **output** frequency is just above the first harmonic of the oscillator in this example.
9. Usually, it is better to select Linear or log, and define your own sweep. Auto usually takes more frequency point than are needed, thus extending the pxf runtime.
10. Set *Maximum sideband* to the same value used for pss harmonics.
11. Select the output net or current, as usual. If you have an output current, add an iprobe in series with the output, and click probe in the *Output* section. Then specify the instance name of the iprobe in the *Output Probe Instance* field.
12. Below the *Sampled* selection, define the output type, and set the output again.
13. Specify an appropriate threshold. This ring oscillator has a differential output, so 0 volts is an appropriate threshold.
14. If desired, set the crossing direction. This example calculates all the crossings of 1.65 volts.

© 1994-2020 All Rights Reserved.

15. Click *OK* in the *Choosing Analyses* form and run the simulation. When the simulation completes, select *Results* - *Direct Plot* - *Main Form*. The *Direct Plot Form* appears.

© 1994-2020
All Rights Reserved.

16. Plot the pss time-domain output waveform(s) so you can see the times of the transitions .

17. For the V4M signal, the rising edge is about 184psec, and the falling edge is about 445psec.

18. Now plot the pxf sampled results.

© 1994-2020
All Rights Reserved.

19. If desired, select *dB20*.

20. Select the event time.

© 1994-2020
All Rights Reserved.

21. Now select the power supply source in the schematic. The results plot, as shown below.

22. To read the result, go to the input frequency on the pxf plot, and read the gain. This is the voltage gain from the power supply to the output just above the first harmonic of the oscillator at a single timepoint in the waveform. If there is voltage gain from the power

© 1994-2020 All Rights Reserved.

supply to the output, it shows up as ripple on the output, which will affect the timing of the output crossing. The gain at 1MHz input frequency is shown below.

© 1994-2020
All Rights Reserved.

Periodic Stability Analysis (PSTB)

Pstb is provided to measure the small-signal loop gain in the presence of a large signal. As an example, think of an amplifier. When the input signal is small, the amplifier is largely linear, and stb and pstb should agree. However, when the input signal becomes large, the amplifier becomes more nonlinear. This will change the loop gain of the feedback system. Pstb measures the loop gain with the large signal present in the system.

An additional application is for measuring the loop gain of oscillators. In most oscillators, the power supplies define the output level. The nonlinearities created by the oscillation cause the loop gain to shift slightly from the linear stb analysis results. We have seen cases where multiple stable oscillation frequencies resulted from the nonlinearities introduced by the oscillations themselves, and this showed up in the pstb result. The large signal oscillation is solved in the pss analysis, and the small-signal stability analysis is run in the pstb analysis.

© 1994-2020 All Rights Reserved.

Example: Variation in Loop Gain of an Opamp with Swept Input Signal Amplitude

Consider the opamp circuit below.

To use pstb, an `iprobe` from `analogLib` (single ended) or a `diffstbprobe` from `analogLib` (differential) needs to be added in series with the feedback path. Note the arrow in the current probe is in the same direction as the direction of the feedback.

The input source to the circuit has the frequency and amplitude set to variable names in order to allow changing the test setup without needing to change the circuit. This is suggested for all the sources used in the circuit that might need to be changed. This allows changes in frequency or amplitude from the ADE Explorer environment without needing to change the schematic itself.

© 1994-2020 All Rights Reserved.

The property list for the input source is shown below.

The easiest way to get the variables list into the ADE Explorer is to select *Variables -> Copy From Cellview*. To set the values, select the value field to the right of the variable, type in the desired value, and press *Enter.* When you have set the variables, if you want to add the

© 1994-2020 All Rights Reserved.

values back into the cellview, in the ADE Explorer window, select *Variables - Copy To Cellview*, and then perform a *Check and Save* in the schematic.

The input frequency for this example is 100KHz and its amplitude is 100 millivolts peak.

The basic strategy is to apply the signal that causes the distortion in the pss analysis, and then measure the loop gain using the pstb analysis.

The pss analysis needs to be set up first. The example below shows a single input at 100KHz with 10 harmonics. The amplitude is swept from 0.1 volts to 1.5 volts peak with a step value of 0.2 volts. For more information, see the pss section at the beginning of this chapter. This

© 1994-2020 All Rights Reserved.

will characterize the amplifier at a range of different input amplitudes. At each amplitude, both the pss and pstb analyses are run.

© 1994-2020 All Rights Reserved.

© 1994-2020
All Rights Reserved.

Next, the pstb analysis needs to be set up. The frequency range depends on your application. The *Probe Instance* is the iprobe from `analogLib` in the circuit. If you have a differential circuit, use `diffstbprobe` from `analogLib`. `diffstbprobe` allows the measurement of common-mode and differential-mode loop gain.

Once the analyses are set up, the simulation can be run. When the simulation finishes, select *Results - Direct Plot - Main Form* in the ADE Explorer window.

© 1994-2020
All Rights Reserved.

The example below shows the *Direct Plot Form*. If you want dB for the loop gain amplitude plot, plot the magnitude and phase separately This example shows dB for the loop gain magnitude plot.

If you select *Add To Outputs*, an expression is entered under the *Maestro* tab of ADE Explorer so that the next time you simulate your circuit, this measurement plots automatically.

© 1994-2020
All Rights Reserved.

Here is the output voltage of the opamp versus input voltage for the sweep. The curve begins to compress at 1.1 volts peak.

© 1994-2020
All Rights Reserved.

Here is the loop gain and phase curve for the 0.1 volt input.

Note that the DC phase is 180, and not zero. From feedback theory, we would expect zero. So what causes this difference? In feedback theory, we always assume there is a forward and reverse transfer function, and an inverting summing junction with a gain of unity. In feedback theory, we plot the forward and reverse transfer functions only. When the phase is zero, and then applied through the inverting input, the phase becomes 180. In the circuit, we cannot separate out the summing junction. Thus, the condition of instability is a linear magnitude of 1 and a phase of zero, or multiples of 360 degrees.

Vertical marker V2 has been placed at (nearly) unity gain (0dB). The phase is about 87.8 degrees. The distance to zero phase is 88 degrees. This is the phase margin for this circuit at low input amplitude. The unity gain frequency is about 800KHz

© 1994-2020 All Rights Reserved.

Below is the plot with a 1.1 volt input.

The DC gain has dropped from about 66dB to about 42 dB. The unity gain frequency has dropped from about 800KHz to about 112KHz. The phase margin has actually improved.

© 1994-2020
All Rights Reserved.

Overview of Simulation Capabilities

Pstb is used to measure the loop gain with a large signal applied to the circuit. First, the large signal is applied in pss. An `iprobe` (single-ended) or a `diffstbprobe` (differential) is placed in the circuit in series with the feedback path. Both probes are in `analogLib`. Pstb applies a voltage to the source(s) in the probe, and measures the amplitude and phase of the signal across the probe. This is the loop gain of the circuit. `diffstbprobe` allows the measurement of common-mode and differential-mode gain in the differential circuit.

Pstb is supported for both shooting and harmonic balance pss.

In some cases, placing the probe in the circuit can break some local feedback paths. If this happens, the loop gain calculation will be incorrect. If you are unsure if this could be happening in your circuit, place the probe in a different position in the loop. If the loop gain stays the same, both results are correct. If it changed, try a third point in the circuit. When the two points agree, you have the loop gain.

Setting Harmonics and Sidebands

Shooting

The pss analysis, when shooting is selected, has a minimum of 200 timepoints, so it inherently has frequency domain content through the 100th harmonic of the pss beat frequency. Maximum harmonic is a pss post-processing property that specifies how many harmonics to calculate in the Fourier transform of the waveform in pss. If the number of harmonics is zero, it just means that no frequency domain information should be calculated in the simulation. If the number of harmonics is 10 or less, there is no effect on the PSS waveform. When the maximum harmonics gets larger than 10, the minimum number of timepoints in the pss analysis is increased. 20 timepoints in the period of the highest harmonic are forced. Thus, if the maximum harmonics is raised, the pss waveform gets more timepoints which improves the accuracy of the waveform at the cost of longer runtimes and more memory required to run the pss simulation.

In order for the pstb analysis to correctly measure conversion gain, the wave shape at the highest analysis frequency needs to be accurate, so with the default settings in shooting pss, up to 40 times the pss beat frequency can be used as the maximum frequency in pstb. If you need a higher stop frequency, either set the number of harmonics to maximum frequency in pstb/4*pss beat frequency, or set the pss option *maxacfreq* to the stop frequency in pstb.

© 1994-2020 All Rights Reserved.

Harmonic Balance

When *Harmonic Balance* is selected in the pss analysis, only the harmonics specified in the pss *Choosing Analyses* form exist in the solution. In the pstb analysis, because only the harmonics specified in the pss-hb *Choosing Analyses* form exist, the maximum frequency term can only be as high as the highest frequency harmonic specified in the pss analysis. If you need a higher maximum frequency in pstb, you need to specify more harmonics in the pss analysis.

© 1994-2020 All Rights Reserved.

ADE Implementation

This section jumps from the theoretical to the practical with examples of how to use pstb. The focus is only on the pstb *Choosing Analyses* form so you can see where the individual settings are made. For examples, with all the steps shown, see the examples section that follows this section.

PSTB Setup

Here is an example of the Choosing Analyses form for a normal loop gain measurement
.

1. At the top of the form the PSS beat frequency of 100KHz is shown.

© 1994-2020
All Rights Reserved.

2. The frequency sweep range specifies the limits and spacing of the loop gain measurement. Note that with this maximum frequency of 100M, either the pss option *maxacfreq* needs to be set to 100M, or the number of pss harmonics needs to be set to 250.

Tip

Maxacfreq forces 5 timepoints in the period of the frequency specified for maxacfreq. PSS harmonics above 10 forces 20 timepoints in the period of the highest harmonic. In this example, the pss beat frequency is 100K, and the highest frequency in pstb is 100M, or a ratio of 1000. Since pss harmonics force four times as many timepoints as maxacfreq does, to get equivalent runs, you need 1000/4 or 250 harmonics in pss.

3. Linear and log sweeps are provided. Generally, log sweeps are used. The number of points depends on how rapidly the loop gain changes with frequency in your circuit. High Q circuits will need more resolution than shown in this example.

4. A probe must be used in the circuit and specified in the *Probe Instance* field. For single-ended circuits, an `iprobe` from `analogLib` should be used. For differential circuits, a `diffstbprobe` should be used. When `diffstbprobe` is used, you will have an additional choice in the *Choosing Analyses* form that allows the selection of differential or common-mode measurements of loop gain.

© 1994-2020 All Rights Reserved.

PSTB Options

Below is the pstb options form.

Solver

The solver option is used only for driven circuits. The default solver is the *turbo* solver.

When hb is used as the engine in pss, leave this option at the default.

© 1994-2020 All Rights Reserved.

When shooting is used, sometimes when the pstb analysis frequency is very close to the frequency of one of the harmonics in the pss result, warning messages will appear in the pstb output warning that the accuracy might not be good enough. If you see these messages, select the *std* solver, which has better ability to handle frequencies that are very close to a harmonic in the pss, but which takes longer to run than the turbo solver.

Oscsolver

The *oscsolver* option is used only for oscillators. The default solver is the *turbo* solver.

When hb is used as the engine in pss, leave this option at the default.

When shooting is used, sometimes when the pstb analysis frequency is very close to the frequency of one of the harmonics in the pss result, warning messages appear in the pstb output warning that the accuracy might not be good enough. If you see these messages, select the *std* solver, which has better ability to handle frequencies that are very close to a harmonic in the pss, but takes longer to run than the *turbo* solver.

For very large oscillator designs, the ira oscsolver may take less memory and run faster with no loss in accuracy, however, it is less robust for convergence compared to *turbo* or *std*.

Hbprecond_solver

This option is only available only when harmonic balance selected in PSS and APS is used.

The basic solver is the only solver available in standard Spectre when harmonic balance is selected. *autoset* is the default solver in harmonic balance when APS is used. This solver is faster, but occasionally stagnates. When stagnation is detected, APS automatically switches to the basic solver, and prints a message in the Spectre output window. If you have stagnation, it will save a small amount of time to set this option to *basicsolver*.

The following options are not commonly used:

Tolerance

Leave this option at the default value.

Pstb uses an iterative solver to calculate the loop gain at the specified frequencies. Any iterative solver needs an error tolerance to specify when to stop iterating because the solution is accurate enough. The tolerance option specifies that accuracy for pstb. For shooting, the

© 1994-2020 All Rights Reserved.

default tolerance is 1e-9. For driven circuits where HB is the pss engine, the default is 1e-6. For oscillators where hb is selected for the pss engine, the default is 1e-4.

Gear_order

Do not change this option.

Resgmrescycle

Leave this option at the default.

Annotate

This option controls the level of detail in the Spectre output log. Selections to the left in the form produce less detail. Selections to the right in the form add increasing detail.

Save

This option is not implemented at the current time.

Nestlvl (Shooting and Harmonic Balance)

This option is not implemented at the current time.

AdditionalParams

In some cases, when shooting pss is used for an oscillator circuit, pstb results near a pss harmonic may be questionable. To solve this issue, type osc_version=dts in the additionalParams field.

additionalParams is typically used for new features that are being beta tested. `keyword=value` pairs are expected in this field.

For more information about the other options, type `spectre -h pstb` at the command prompt in a Unix shell window.

© 1994-2020 All Rights Reserved.

Oscillator Example

For the differential oscillator circuit below, a `diffstbprobe` has been placed at mirror locations in the differential circuit. This is shown in the red box below.

© 1994-2020
All Rights Reserved.

1. First, a pss analysis needs to be set up. For more information, refer to the pss section at the beginning of this chapter.

© 1994-2020
All Rights Reserved.

2. Now set up the pstb analysis. The resolution of this sweep is quite high.

The frequency sweep range, and the number of points will be determined by your circuit. In this case, from linear stb analysis, all the interesting variations started at 1GHz. Narrow peaks were detected, which forced many points in order to resolve the amplitude of the peaks.

3. Now run the simulation. When the simulation completes, select *Results - Direct Plot - Main Form* in the ADE Explorer window.

© 1994-2020
All Rights Reserved.

4. Now the loop gain and phase are plotted using the *Direct Plot Form*.

© 1994-2020
All Rights Reserved.

5. Now, the plot of the loop gain and phase is shown. The red trace is loop gain, and the blue trace is the phase.

6. A vertical marker has been placed at the zero degree point on the phase curve. Note that the gain is very close to unity, and the phase is very close to zero. Remember that in pstb, the summing junction is included in the analysis, so the condition for oscillation is unity gain and zero or multiples of 360 degrees of phase. The frequency readout shows 5.1499GHz.

7. For an oscillator, you should see almost exactly unity gain and a phase of almost exactly zero or multiples of 360 degrees. Small differences are to be expected because of finite resolution in the frequency sweep and/or interpolation error in the markers.

© 1994-2020 All Rights Reserved.

8. The plot of the output spectrum is shown below.

Note that the frequency in the large-signal pss is the same as the small-signal pstb analysis. The nonlinearities of the oscillator are taken into account in the pstb simulation.

© 1994-2020
All Rights Reserved.

Periodic S-parameter Analysis (PSP)

Psp is provided to measure S-Parameters for circuits that translate frequencies. All the basic assumptions of S-Parameters apply to psp, with the addition of having different input and output frequencies. Psp can be used for up or down-conversion applications.

Example: Down-Conversion Diode Mixer

Consider the double-balanced diode mixer circuit below.

The input source has the frequency and amplitude set to variable names in order to allow changing the test setup without needing to change the circuit. This is suggested for all the sources used in the circuit that might need to be changed. This allows changes in frequency or amplitude from the ADE Explorer environment without needing to change the schematic itself.

© 1994-2020 All Rights Reserved.

The property list for the RF input port is shown below.

© 1994-2020
All Rights Reserved.

The same strategy is used for the LO port.

The easiest way to get the variables list into ADE Explorer is to select *Variables -> Copy From Cellview*. To set the values, select the value field to the right of the variable, type in the desired value, and press *Enter.* When you have set the variables, if you want to add the values back into the cellview, in the ADE Explorer window, select *Variables - Copy To Cellview*, and then perform a *Check and Save* in the schematic.

The RF input frequency is disabled (by setting the frequency to zero), and the LO is at 1GHz with a 1.4 volt peak signal. This is +13 dBm, which is a common drive level for a diode mixer.

The basic strategy is to apply the signal that causes the frequency translation in the pss analysis, and then measure the frequency-shifted S-Parameters using psp. Doing this allows measuring the input match at the input frequency, and the output match at the output frequency in the same analysis. For the forward and reverse gain terms, the specified frequency translation is taken into account.

© 1994-2020 All Rights Reserved.

The pss analysis needs to be set up first. The example below shows a single input at 1GHz with 10 harmonics. For more information, see the pss section at the beginning of this chapter.

© 1994-2020
All Rights Reserved.

Next, the psp analysis needs to be set up. The frequency range depends on your application, and can be either an input frequency range, or an output frequency range. The actual frequency for each port-frequency definition will be set up later. In this example, the frequency range is just above the LO frequency at 1GHz.

© 1994-2020 All Rights Reserved.

Start with Port 1. In the *Port #* column, type 1. In the *Name* column, either type the instance name, or use the *Select Port* button just below the edit line to select the port in the circuit. In this case, Port 1 is the RF port.

© 1994-2020
All Rights Reserved.

Now click *Choose Harmonic*. In the *Choose Harmonic* window, select the frequency range you want associated with that port number. For the down-conversion application, The input is just above the LO frequency of 1GHz.

Click OK.

© 1994-2020 All Rights Reserved.

Click *Add* located to the right of the *Choose Harmonic* button. This adds that port and frequency definition to the list for the analysis. By convention, Port 1 defines the input frequency.

Note that Port 1 has both a port and a frequency definition associated with it. In psp, both are necessary for a port definition.

© 1994-2020
All Rights Reserved.

In psp, multiple port-frequency definitions can be run at one time.

The next step is to measure RF to IF conversion gain. Type 2 in the *Port #* column. Click *Select Port*, and select the IF port.

© 1994-2020
All Rights Reserved.

Now define the frequency range for that port. Click *Choose Harmonic*, and select the IF frequency from the list.

Click OK.

© 1994-2020 All Rights Reserved.

Click *Add* located to the right of the *Choose Harmonic* button. The second port-frequency definition is added to the list. By convention, Port 2 defines the output frequency range.

© 1994-2020
All Rights Reserved.

Now measure the RF to IF isolation. Type 3 in the *Port #* column. Click *Select Port*, and again select the IF output port. Click *Choose Harmonic*, and select the input frequency range.

Click OK.

© 1994-2020 All Rights Reserved.

Click *Add* located to the right of the *Choose Harmonic* button. The third port-frequency definition is added.

© 1994-2020
All Rights Reserved.

If a noise simulation is desired, set the *Maximum sideband* field to the same number of harmonics used in the pss analysis.

With the above port-frequency definitions, S_{11} measures the input match at the RF frequency. S_{22} measures the output match at the IF frequency. S_{31} measures the RF to IF isolation, and s_{21} measures the RF to IF conversion gain.

© 1994-2020
All Rights Reserved.

Once the analyses are set up, the simulation can be run. When the simulation finishes, select *Results - Direct Plot - Main Form* in the ADE Explorer window.

The *Direct Plot Form* is displayed, as shown below.

© 1994-2020
All Rights Reserved.

Note: Many of the measurements in the *Direct Plot Form*, like Kf, and B1f are only defined for two-port networks. In the *Direct Plot Form*, if you select one of those items for plotting, you will see a message that says the data is not 2 X 2. Also note that Kf and B1f are only defined for amplifiers where there is no frequency translation. If you run a two-port simulation on a mixer like this, and you select Kf to be plotted, you will see a message that the harmonic of the two ports are not the same, and as a result, this cannot be plotted.

If you click *Add To Outputs*, an expression is entered under the *Maestro* tab of the ADE Explorer window so that the next time you simulate your circuit, this measurement plots automatically.

Here are the S11, S21, S31, and S33 plots.

The conversion loss is about 6.5 dB. The input match is about -7.9dB. The output match is about -9.8dB. The RF to IF isolation is about 84 dB.

Note that these measurements can be plotted on a Smith Chart as well. The Smith Chart capability is only available from the *Direct Plot Form*. under the *Maestro* tab of the ADE Explorer window will always plot on a rectangular plot. The Smith Chart has not been shown here because this circuit is inherently very wide band. The curve on the Smith Chart would be tiny. An example using a Smith Chart is shown later on in the Examples section.

© 1994-2020 All Rights Reserved.

Psp can be used for up conversion as well. To do that, just change the frequency definitions for the ports. Since IF to RF gain is usually not needed, only a two-port measurement has been set up. Note the frequency difference for Ports 1 and 2.

© 1994-2020
All Rights Reserved.

Now run the analysis. S11, S21, and S22 are plotted, as shown below.

© 1994-2020
All Rights Reserved.

Overview of Simulation Capabilities

Psp is similar to sp in that power gain measurements are made. Sp is based on the DC operating point, so it can only produce results that are linear. All the frequencies are the same for all the S-Parameter measurements in the sp analysis.

Psp uses the pss analysis results as the basis for making the S-Parameter measurement. Because the PSS result is the operating point solution for psp, the input can mix with any of the harmonics that are present in the pss analysis for the frequency-translated terms. The term SXX is not enough to describe the system. We need not only port numbers, but also the frequency needs to be defined for each port definition.

In the example in the preceding section, the LO was at 1GHz and the specified frequency was just above the first harmonic of the LO. In the psp definition for each port, both the port name and the frequency that is associated with that measurement need to be defined.

Note: Some measurements like Kf are only defined for systems that do not translate frequency. In psp, because the calculation is based on the pss large-signal solution, you can sweep the input power to an amplifier, and measure the stability factor as it changes with the input amplitude. In this case, the input and output frequencies are the same, so Kf is defined. If you have different frequencies on the input and output, Kf cannot be plotted because it is not defined. The *Direct Plot Form* will say that term cannot be plotted. Otherwise, all the parameters that can be plotted in sp are available in psp. Also, note that Kf and B1f are only defined for 2-port networks.

Noise analysis is available in psp, and is the same as pnoise. Please see the pnoise chapter for noise information.

Setting Harmonics and Sidebands

Shooting

The pss analysis, when shooting is selected, has a minimum of 200 timepoints, so it inherently has the frequency domain content through the 100th harmonic of the pss beat frequency. Maximum harmonic is a pss post-processing property that specifies how many harmonics to calculate in the Fourier transform of the waveform in pss. If the number of harmonics is zero, it just means that no frequency domain information should be calculated in the simulation. If the number of harmonics is 10 or less, there is no effect on the PSS waveform. When maximum harmonics gets larger than 10, the minimum number of timepoints in the pss analysis is increased. 20 timepoints in the period of the highest harmonic are forced. Thus, if maximum harmonics is raised, the pss waveform gets more timepoints which increases the accuracy of the pss waveform at the cost of longer runtimes and more memory required to run the pss simulation.

© 1994-2020 All Rights Reserved.

In order for the psp analysis to correctly measure conversion gain, the wave shape at the highest analysis frequency needs to be accurate, so with the default settings in shooting pss, up to 40 times the pss beat frequency can be used as the maximum frequency in psp. This will cover the vast majority of cases.

Harmonic Balance

When *Harmonic Balance* is selected in the pss analysis, only the harmonics specified in the pss *Choosing Analyses* form exist in the solution. If the harmonic balance analysis has enough harmonics to accurately capture the behavior of your circuit, the psp result will be accurate. To see if the result is accurate, increase the number of pss harmonics by about 50% and run again. If the psp result did not change, then you had enough harmonics to begin with, and might be able to be reduced. If it did change, add more harmonics and run again. For the fastest simulation, use the smallest number of harmonics that produce a stable psp result.

In the psp analysis, because only the harmonics specified in the pss-hb *Choosing Analyses* form exist, the maximum frequency term can only be as high as the number of harmonics specified in the pss analysis times the pss beat frequency.

© 1994-2020 All Rights Reserved.

ADE Implementation

This section jumps from the theoretical to the practical with examples of how to use psp. The focus is only on the psp *Choosing Analyses* form so you can see where the individual settings are made. For examples with all the steps shown, see the examples section that follows this section.

PSP Setup

For this example, consider an LNA-Mixer circuit shown below.

© 1994-2020
All Rights Reserved.

1. First, define either an input or output frequency range in the frequency sweep range fields. It is usually easier to define an input frequency range, as shown below for a 2.4GHz application.

2. The frequency sweep range specifies the limits and spacing for the input frequency.

© 1994-2020
All Rights Reserved.

3. Linear and log sweeps are provided. It is usually better to select linear or log, and specify the number of steps instead of using automatic. Automatic will run 50 frequency points, which usually is not needed for most applications.

4. Now port and frequency definitions need to be made.

5. In the text field just above the *Select Port* button, type 1 in the *Port #* field. Click *Select Port*, and select the input port.

© 1994-2020
All Rights Reserved.

6. Click *Choose harmonic*. In the window that appears, select the input frequency range.

7. Click *OK*.

© 1994-2020 All Rights Reserved.

8. Click *Add*. The port-frequency combination is added to the list. By convention, Port 1 sets the input frequency for psp.

© 1994-2020
All Rights Reserved.

9. Now define the output port-frequency combination as Port 2.

© 1994-2020
All Rights Reserved.

10. Repeat the process to define Port 3 as the gain from RF to IF with no frequency translation. (The RF to IF isolation.)

11. Click *Add* to add the last port-frequency definition to the list.

12. If you want a noise simulation, set *Maximum sideband* to the number of LO or clock harmonics. This analysis is the same as pnoise. For more information, see the pnoise section in this chapter.

© 1994-2020 All Rights Reserved.

PSP Options

Below is the psp options form.

None of the options are commonly used.

© 1994-2020 All Rights Reserved.

Solver

The solver option is used only for driven circuits. The default solver is the *turbo* solver.

When hb is used as the engine in pss, leave this option at the default.

When shooting is used, sometimes when the psp analysis frequency is very close to the frequency of one of the harmonics in the pss, warning messages will appear in the psp output warning that the accuracy might not be good enough. If you see these messages, select the *std* solver, which has better ability to handle frequencies that are very close to a harmonic in the pss, but which takes longer to run than the *turbo* solver.

Oscsolver

The *oscsolver* option is used only for oscillators. The default solver is the *turbo* solver.

When hb is used as the engine in pss, leave this option at the default.

When shooting is used, sometimes when the psp analysis frequency is very close to the frequency of one of the harmonics in the pss, warning messages appear in the psp output warning that the accuracy might not be good enough. If you see these messages, select the *std* solver, which has better ability to handle frequencies that are very close to a harmonic in the pss, but takes longer to run than the *turbo* solver.

For very large oscillator designs, the ira oscsolver may take less memory and run faster with no loss in accuracy, however, it is less robust for convergence compared to *turbo* or *std*.

Hbprecond_solver

This option is only available only when harmonic balance is selected in PSS and APS is used.

The basic solver is the only solver available in standard Spectre when harmonic balance is selected. *autoset* is the default solver in harmonic balance when APS is used. This solver is faster, but occasionally stagnates. When stagnation is detected, APS automatically switches to the basic solver, and prints a message in the Spectre output window. If you have stagnation, it will save a small amount of time to set this option to *basicsolver*.

Tolerance

Leave this option at the default value.

© 1994-2020 All Rights Reserved.

Psp uses an iterative solver to calculate the output amplitudes. Any iterative solver needs an error tolerance to specify when to stop iterating because the solution is accurate enough. The tolerance option specifies that accuracy for psp. For shooting, the default tolerance is 1e-9. For driven circuits where HB is the pss engine, the default is 1e-6. For oscillators where hb is selected for the pss engine, the default is 1e-4.

Gear_order

Do not change this option.

Resgmrescycle

Leave this option at the default. For the *resgmres* linear solver, there are several different options.

Annotate

This option controls the level of detail in the Spectre output log. Selections to the left in the form have less detail in the Spectre output log. Selections to the right in the form have increasing levels of detail in the output log.

Freqaxis

Psp displays all the results on the same frequency scale. The default is *absin* (absolute value of the input frequency) If there is a negative input frequency, *in* will show that frequency on the negative frequency axis. *out* shows the output frequency range on the X axis.

AdditionalParams

additionalParams is typically used for new features that are being beta tested. Entries in this field are `keyword=value` pairs.

For more information about the other options, type `spectre -h psp` at the command prompt in a Unix shell window.

© 1994-2020 All Rights Reserved.

Examples

For this example, consider an LNA-Mixer circuit shown below.

© 1994-2020
All Rights Reserved.

1. Set up the pss analysis. For more detail, see the pss section at the beginning of this chapter.

© 1994-2020
All Rights Reserved.

2. In the psp *Choosing Analyses* form, define either an input or output frequency range in the frequency sweep range fields. It is usually easier to define an input frequency range, as shown below for a 2.4GHz application.

3. It is usually easier to define the frequency sweep range to be the input frequency range.

© 1994-2020
All Rights Reserved.

4. Linear and log sweeps are provided. It is usually better to select linear or log, and specify the number of steps instead of using automatic. Automatic will run 50 frequency points, which usually is not needed for most applications.

5. Now port and frequency definitions need to be made.

6. In the field just above the *Select Port* button, type 1 in the *Port #* field. Click *Select Port*, and select the input port.

© 1994-2020
All Rights Reserved.

7. Click *Choose harmonic*. In the window that appears, select the input frequency range. and click OK.

© 1994-2020 All Rights Reserved.

8. Click *Add*. The port-frequency combination is added to the list. By convention, Port 1 defines the input frequency range for the psp analysis.

© 1994-2020
All Rights Reserved.

9. Now define the output port-frequency combination as Port 2.

© 1994-2020
All Rights Reserved.

10. Note that the frequency was selected by clicking *Choose Harmonic*, and selecting the output frequency range as shown below.

© 1994-2020 All Rights Reserved.

11. Next, click *Add*. This adds the port-frequency combination as Port 2 in the list. By convention, Port 2 defines the output frequency range for the psp analysis

© 1994-2020
All Rights Reserved.

12. Repeat the process to define Port 3 as the gain from RF to IF with no frequency translation. (The RF to IF isolation).

13. If you want a noise simulation, set *Maximum sideband* to the same value that was used for the LO or clock harmonics in the pss analysis. This is the same analysis as pnoise. For more information about pnoise, see the pnoise section in this chapter.

© 1994-2020 All Rights Reserved.

14. Now run the simulation. When the simulation completes, select *Results - Direct Plot - Main Form* in the ADE Explorer window.

© 1994-2020
All Rights Reserved.

15. Select *psp* from the *Analysis* section.

16. To plot S-Parameters, select *SP* from the *Function* section. In this case, Port 1 is the input port at the RF frequency, Port 2 is the output port at the IF frequency, and Port 3 is the output port at the RF frequency.

17. In the *Plot Type* section, several choices are available. The input match (S11) can be plotted on the Smith Chart. Select *Z-Smith* from the *Plot Type* section.

18. Now, click the *S11* button. S11 appears on the Smith Chart.

The length of the vector from the center of the circle to the curve is the magnitude of the reflection coefficient. In the above plot, the input match becomes better at high frequency because the vector is shorter. Most LNAs are deliberately mismatched because a lower noise figure can typically be obtained with a small mismatch.

19. There are two useful zoom functions for the Smith Chart. typing the f key fits the actual data to the Smith Chart. <Shift>-F fits the unit circle, as shown above.

© 1994-2020
All Rights Reserved.

20. Now plot S21 on the Smith Chart. It displays outside of the unit circle.

© 1994-2020
All Rights Reserved.

21. Now plot S11, S21, S22, and S31 on a rectangular plot. The conversion gain is about 19.1dB. The input match is questionable. The output match is not good, but it does not matter because the output is at low frequency. The RF to IF isolation is about 84dB.

22. When you select Kf, because this is a 3-port analysis, the *Direct Plot Form* shows that Kf can not be plotted. Kf is defined for a 2-port simulation only. If you removed the RF to IF S-parameter measurement in this example, and converted to a 2-port simulation, you

© 1994-2020
All Rights Reserved.

still would not be able to plot the data. Kf and B1f are defined only for amplifiers where the input and output frequency are the same.

© 1994-2020
All Rights Reserved.

Swept Input Power In a PA

For the next example, consider a power amplifier.

© 1994-2020
All Rights Reserved.

1. First, set up the pss analysis. In this case, the input power is swept. For more information, see the pss section at the beginning of this chapter.

© 1994-2020 All Rights Reserved.

2. Now define the input frequency and input port for psp. When a sweep is run in pss, the psp frequency sweep will switch to single point automatically.

© 1994-2020
All Rights Reserved.

3. Click *Choose Harmonic*. In the window that appears, select the input frequency and click *OK*.

© 1994-2020 All Rights Reserved.

4. Click *Add*. The port/frequency combination is entered in the list.

© 1994-2020
All Rights Reserved.

5. Repeat for the output port/frequency combination.

© 1994-2020
All Rights Reserved.

6. If you want a noise analysis, set *Maximum sideband* to the same value used for pss harmonics. This analysis is the same as pnoise. Please refer to the pnoise section earlier in this chapter for details.

© 1994-2020
All Rights Reserved.

7. Click *OK*, and run the simulation. When the simulation completes, select *Results - Direct Plot - Main Form* in ADE Explorer. The *Direct Plot Form* is displayed.

8. Select *pss* from the *Analysis* section.

9. Plot the compression curve for the amplifier.

© 1994-2020 All Rights Reserved.

10. The compression point is shown below.

© 1994-2020
All Rights Reserved.

11. Now plot stability factor from the psp results (Kf).

© 1994-2020
All Rights Reserved.

12. The plot displayed, as shown below.

13. To make the plot more usable, move your mouse cursor over one of the numbers on the X Axis, and click the right mouse button and select *Swap Sweep Var* from the context menu.

© 1994-2020
All Rights Reserved.

14. A new window appears. Type the frequency value in the X Value field and click OK.

Kf is plotted. The amplifier is stable when the gain is greater than one and the stability factor is greater than one.

© 1994-2020
All Rights Reserved.

B1f is also available. When B1f is greater than zero, and the gain is greater than one, the system is stable.

Both plots show increasing stability as the signal amplitude gets larger.

© 1994-2020
All Rights Reserved.

© 1994-2020 All Rights Reserved.

5

Multiple Input Large and Small-signal Analyses

Quasi-Periodic Steady-State Analysis (QPSS)

QPSS is a large-signal analysis that is intended for two or more input frequencies. Both shooting and harmonic balance engines are available in qpss. In general, harmonic balance is faster and uses less memory than shooting for most applications. Shooting qpss should be used for the cases where harmonic balance just does not work because the system is too nonlinear. Examples include sampling circuits with a sample clock and a second periodic input, and switched-capacitor circuits with a sample clock and a periodic input.

In general, when harmonic balance qpss is needed, it is better to use the hb analysis as described in Chapter 3 of this user guide. The reason for this suggestion is that all the harmonic balance improvements are being implemented in the hb analysis *Choosing Analyses* form, and generally are not added to the qpss *Choosing Analyses* form. An example is the ability to run an oscillator and a periodic input, which can be done in hb, but not in qpss. Another example is auto harmonic selection.

As a strong recommendation, when you need harmonic balance qpss, switch to the hb *Choosing Analyses* form instead.

© 1994-2020 All Rights Reserved.

Example

Consider a sampling circuit shown below.

© 1994-2020
All Rights Reserved.

The input source is sinusoidal, and the frequency and amplitude have been set to variable names.

This allows the setting of the frequency and amplitude by changing those variables in the ADE Explorer environment without needing to change the circuit. Note also that there is a DC level

© 1994-2020
All Rights Reserved.

for the sinusoid. This is because the circuit only includes an N FET switch. This places the signal in the middle of the range of the circuit.

Variables are also used to set the sample clock.

© 1994-2020
All Rights Reserved.

Those variables are set in ADE Explorer.

For this example, the input frequency is 1MHz with a 50mV peak amplitude. The sample clock is 100MHz.

© 1994-2020 All Rights Reserved.

Now the qpss analysis needs to be set up.

The *Engine* is *shooting*. One signal must be set to *Large*, and the rest must be set to *Moderate*. The large signal should be the one that causes the most distortion in the circuit. In this case, that signal is the sample clock. *Large* and *Moderate* are just terminologies used

© 1994-2020 All Rights Reserved.

by qpss. The qpss solution is a large-signal solution with all the signals applied to the circuit. More information will be provided later on in this chapter.

The number of harmonics to be calculated must also be set. For reasons described later, the number of harmonics on the Large signal are not critical. For the moderate signals, if the signal amplitude is well below the compression point, two or three harmonics are suggested. If a signal is at the compression point, five harmonics are suggested. If the signal is above the compression point, start with seven harmonics. You may need more harmonics to get an accurate solution.

Now run the simulation. When the simulation completes, select *Results - Direct Plot - Main Form* in ADE Explorer. The *Direct Plot Form* is displayed.

© 1994-2020
All Rights Reserved.

Select qpss results. In this case, the output spectrum will be displayed. Peak is selected because the input is 50mV peak. This allows a direct comparison to be made. Magnitude and not dB is set to get a direct comparison.

Now select the net in the circuit. The output net is selected for this example.

The waveform is displayed.

The waveform is zoomed in near DC. A marker has been placed to measure the output amplitude at 1MHz. As expected, the output amplitude is slightly less than 50mV peak.

© 1994-2020 All Rights Reserved.

The sample clock feedthrough is also measured. In the zoomed in view near 100MHz (the sample frequency), the amplitude is about 62 mV peak

.

QPSS Shooting Concepts

Think of an example where 1GHz and 1GHz plus 1MHz frequencies are applied to a circuit. If we try to use PSS Shooting on this case, then 1000 cycles of 1GHz and 1001 cycles of 1.001GHz need to be simulated by solving for the continuous waveform that is produced.

This is impractical for several reasons. First, shooting pss is known to have convergence problems when the total number of cycles of the highest frequency to be simulated exceeds somewhere between 50 and 250 cycles, depending on the nonlinearity of the circuit. Second, when 1000 cycles need to be simulated, at least 20K timepoints are required to get a reasonably accurate waveform. Because PSS Shooting saves not only the solution voltages and currents (if selected) but also the solution matrix at each timepoint, the amount of memory required for a simulation like this is very high.

If the pss simulation could be completed, all the harmonics of 1MHz would need to be calculated through at least the third harmonic of 1001 MHz, or 3003 harmonics. Not only does the time-domain simulation take a long time, but also the Fourier transform takes a long time.

© 1994-2020 All Rights Reserved.

If you have this case where the pss beat frequency is a small fraction of the highest frequency, use qpss or hb instead.

Qpss is similar to harmonic balance in that you need to specify the number of harmonics to calculate for each tone in the circuit. Qpss calculates the harmonics of the inputs and the mixing products that are specified by the number of harmonics that are set for each input tone. It does not calculate all the harmonics of 1 MHz. Because it does not calculate all the harmonics, it does not need all the data that the pss analysis needs, thus it runs faster than PSS Shooting for an equivalent set of input frequencies.

Qpss does not simulate the continuous waveform. Instead, it solves for the waveform in small slices of time during the overall period. The more harmonics you want to solve for requires that more slices of time need to be simulated. Just like in harmonic balance, calculating more harmonics takes more time and memory. All the input signals are applied in the circuit, and qpss solves for the steady-state waveforms in each slice of time. This time-domain output is never available for plotting. Instead, an ifft is offered to calculate the time-domain waveform.

Qpss shooting requires that you select one signal as large, and the rest of the signals are set to moderate. Large and moderate are just terminology used by the qpss analysis. Qpss shooting is a large-signal analysis with all the signals applied.

The large signal should be the signal that causes the largest amount of distortion in the circuit, or the non-sinusoidal input if there is one present. All the moderate tones must be sinusoidal.

The large signal specifies the period of each slice in time. In order for qpss shooting to be efficient, the large signal must be a high frequency so that the period of each slice is kept short. If you take an example of 1MHz and 1GHz being applied to the circuit, you must select the 1GHz signal as large, or the runtime of qpss will be really long. If you take this example, if you simulate slices of 1GHz, everything works. If you simulate slices of 1MHz, you have very long runtimes.

© 1994-2020 All Rights Reserved.

Here is an example for a system with 1GHz and 1.001GHz applied. The overall waveform contains 1000 cycles of the 1GHz signal and 1001 cycles of the 1.001GHz signal. This waveform comes from the transient analysis.

© 1994-2020
All Rights Reserved.

Below are several shots of a 1 nsec period at different times during the simulation. This is at the very beginning of the simulation.

This is 1/4th of the way through. The waveform in this slice is very different.

© 1994-2020
All Rights Reserved.

© 1994-2020
All Rights Reserved.

This is half way through.

© 1994-2020
All Rights Reserved.

This is 3/4ths of the way through.

What qpss shooting does is solve for the steady-state waveform with all the signals applied in the slices that are needed in order to calculate the number of harmonics that are specified for each signal.

Calculating harmonics for the large signal in each slice and averaging the power at each harmonic gives the power in the large signal.

For one moderate signal, (2 * Number of harmonics) + 1 slices need to be simulated. For three harmonics on the moderate signal, then 7 slices need to be simulated. This gives enough information to solve for the positive frequencies, negative frequencies, and the DC term to be calculated.

If there are two moderate signals, then the number of slices is [(2 * Number of harmonics on moderate 1) + 1] * [(2 * Number of harmonics on moderate 2) + 1]. If there are three harmonics on each moderate tone, 7 * 7 or 49 slices need to be simulated. The waveform becomes more complex and the number of harmonics that are created by the system are larger when three tones are applied to the system. below is the transient waveform for 1GHz,

© 1994-2020 All Rights Reserved.

1.004GHz and a 1.005GHz signals applied to the system. This waveform is from the transient analysis.

Each slice is a pss analysis where the solution voltages and currents (if specified) and the solution matrix need to be saved at every timepoint of every pss analysis. For more information about the pss analysis, see the beginning of Chapter 4.

At the end of the simulation, the harmonic content is calculated based on the waveforms in the slices. Since all the input signals are applied in every slice, the harmonics of all the mixing products can be calculated.

© 1994-2020 All Rights Reserved.

What Happens When you Start a QPSS Analysis?

First, two pss iterations are performed with only the large signal applied. This gives an approximate harmonic content with just the large signal applied. This is shown in the Spectre output window, as shown below.

```
/servers/scratch03/kamal/spectrerf131/testcase131/doc
File  Edit  View  Help                                   cādence
    qpss: time = 503.4 ns    (33.6 %), step = 125 ps       (1.25 %)
    qpss: time = 503.9 ns    (38.6 %), step = 125 ps       (1.25 %)
    qpss: time = 504.4 ns    (43.6 %), step = 125 ps       (1.25 %)
    qpss: time = 504.9 ns    (48.6 %), step = 125 ps       (1.25 %)
    qpss: time = 505.3 ns    (53.1 %), step = 103.6 ps     (1.04 %)
    qpss: time = 505.8 ns    (58.1 %), step = 125 ps       (1.25 %)
    qpss: time = 506.3 ns    (63.1 %), step = 125 ps       (1.25 %)
    qpss: time = 506.8 ns    (68.1 %), step = 125 ps       (1.25 %)
    qpss: time = 507.3 ns    (73.1 %), step = 125 ps       (1.25 %)
    qpss: time = 507.8 ns    (78.1 %), step = 125 ps       (1.25 %)
    qpss: time = 508.3 ns    (83.1 %), step = 125 ps       (1.25 %)
    qpss: time = 508.8 ns    (88.1 %), step = 125 ps       (1.25 %)
    qpss: time = 509.3 ns    (93.1 %), step = 125 ps       (1.25 %)
    qpss: time = 509.8 ns    (98.1 %), step = 125 ps       (1.25 %)
Total time required for tstab analysis `qpss': CPU = 475.927 ms, elapsed =
Time accumulated: CPU = 475.927 ms, elapsed = 1.07287 s.
Peak resident memory used = 36 Mbytes.

Conv norm = 61.4, max dV(Out1) = 1.42052 V, took 20 ms.


================================
`qpss': time = (500 ns -> 510 ns)
================================
    qpss: time = 500.3 ns    (2.77 %), step = 33.33 ps     (333 m%)
    qpss: time = 500.8 ns    (7.77 %), step = 33.33 ps     (333 m%)
    qpss: time = 501.3 ns    (12.8 %), step = 33.33 ps     (333 m%)
    qpss: time = 501.8 ns    (17.8 %), step = 33.33 ps     (333 m%)
    qpss: time = 502.3 ns    (22.8 %), step = 33.33 ps     (333 m%)
    qpss: time = 502.8 ns    (27.8 %), step = 33.33 ps     (333 m%)
    qpss: time = 503.3 ns    (32.8 %), step = 33.33 ps     (333 m%)
    qpss: time = 503.8 ns    (37.8 %), step = 33.33 ps     (333 m%)
    qpss: time = 504.3 ns    (42.8 %), step = 33.33 ps     (333 m%)
    qpss: time = 504.8 ns    (47.8 %), step = 33.33 ps     (333 m%)
    qpss: time = 505.3 ns    (52.7 %), step = 33.33 ps     (333 m%)
    qpss: time = 505.8 ns    (57.7 %), step = 33.33 ps     (333 m%)
    qpss: time = 506.3 ns    (62.7 %), step = 33.33 ps     (333 m%)
    qpss: time = 506.8 ns    (67.7 %), step = 33.33 ps     (333 m%)
    qpss: time = 507.3 ns    (72.7 %), step = 33.33 ps     (333 m%)
    qpss: time = 507.8 ns    (77.7 %), step = 33.33 ps     (333 m%)
    qpss: time = 508.3 ns    (82.7 %), step = 33.33 ps     (333 m%)
    qpss: time = 508.8 ns    (87.7 %), step = 33.33 ps     (333 m%)
    qpss: time = 509.3 ns    (92.7 %), step = 33.33 ps     (333 m%)
    qpss: time = 509.8 ns    (97.7 %), step = 33.33 ps     (333 m%)
Conv norm = 8.76e-03, max dV(Out1) = -289.884 uV, took 40 ms.

Number of accepted pss steps =            452

25  >                                                    L1    C1
```


© 1994-2020 All Rights Reserved.

Next, two (by default) stabilization cycles are performed. All the signals are applied to the circuit, and a very simple-minded optimization to attempt to have all the currents sum to zero at all the harmonics at all the nodes is performed. In this phase you will see that all the slices are being simulated. There is only a very small time delay between the first set of slices and the second set of slices. These slices are reported as transient integrations in the qpss output window. In this case, there was a large signal and a single moderate signal with three harmonics specified. This results in seven slices that need to be simulated. Each slice is called a transient integration in the Spectre output window.

© 1994-2020 All Rights Reserved.

The number of stabilization cycles can be controlled by an option called stabcycles. This is shown in the Spectre output window, as shown below.

Finally, a a series of full Newton iterations to the solution of the system is performed. Again, all the slices are solved, and iterating continues until convergence is reached. After each set of slices is simulated, there is a delay while the next iteration is calculated in the Newton

© 1994-2020 All Rights Reserved.

iteration. A maximum of 20 Newton iterations are allowed by default, and can be changed by setting the maxperiods option. This circuit converged in three Newton iterations.

```
/servers/scratch03/kamal/spectrerf131/testcase131/doc
File  Edit  View  Help                                   cadence
......9......8......7......6......5......4......3......2......1......0
3rd_Transient_Integration:

......9......8......7......6......5......4......3......2......1......0
4th_Transient_Integration:

......9......8......7......6......5......4......3......2......1......0
5th_Transient_Integration:

......9......8......7......6......5......4......3......2......1......0
6th_Transient_Integration:

......9......8......7......6......5......4......3......2......1......0
7th_Transient_Integration:

......9......8......7......6......5......4......3......2......1......0
8th_Transient_Integration:

......9......8......7......6......5......4......3......2......1......0
9th_Transient_Integration:

......9......8......7......6......5......4......3......2......1......0
10th_Transient_Integration:

......9......8......7......6......5......4......3......2......1......0
11th_Transient_Integration:

......9......8......7......6......5......4......3......2......1......0
Newton iter  = 1, convNorm = 27.1e-15, maximum dV(In1) = 888.178 aV, took
Number of accepted qpss steps = 123 each in 11 time intervals.
qpss: The steady-state solution was achieved in 3 iterations.
Starting spectrum calculation.
Total time required for qpss analysis `qpss': CPU = 574.913 ms, elapsed =
Time accumulated: CPU = 1.00985 s, elapsed = 2.01587 s.
Peak resident memory used = 37.2 Mbytes.

modelParameter: writing model parameter values to rawfile.

Opening the PSF file ../psf/modelParameter.info ...
element: writing instance parameter values to rawfile.

Opening the PSF file ../psf/element.info ...
outputParameter: writing output parameter values to rawfile.

Opening the PSF file ../psf/outputParameter.info ...
designParamVals: writing netlist parameters to rawfile.

Opening the PSFASCII file ../psf/designParamVals.info ...
primitives: writing primitives to rawfile
25  >                                                    L1    C1
```


© 1994-2020 All Rights Reserved.

ADE Implementation

In this section, just the qpss analysis *Choosing Analyses* form and the options for qpss are shown. An examples section follows with all the steps needed for a simulation.

To open the *Choosing Analyses* form select *Analysis* - *Choose* in ADE Explorer, or select the *Choosing Analyses* icon

located on the upper right of the ADE Explorer window. You can also click the *Click to add analysis* text under the *Analysis* section in the *Setup* pane of ADE Explorer.

© 1994-2020 All Rights Reserved.

The *Choosing Analyses* form is displayed as shown below.

In the *Choosing Analyses* form:

1. Select *qpss*.

© 1994-2020
All Rights Reserved.

2. The default engine is *Shooting*. If you prefer Harmonic Balance, use the hb form, as this form is most up to date.
3. All the sources are listed at the top of the form. To change the settings, select one of the lines, and change the values in the editing line just below the list.
4. One signal must be *Large*. This should be the signal that causes the most distortion in the circuit. The frequency of this signal must be relatively high compared to the highest input frequency, or the simulation will be quite slow and require a lot of memory. If there is a non-sinusoidal source in the circuit, that signal must be set as large. All the moderate tones must be sinusoids.
5. Select an accuracy level. Liberal is not suggested.

Setting errpreset changes the defaults of a number of qpss options. The options that are affected are shown in the table below.

Errpreset	Reltol	Lteratio	Maxstep
`liberal`	`1e-3`	3.5	Period/80
`moderate`	`1e-4`	3.5	Period/100
`conservative`	`1e-5`	10; 3.5 if reltol is smaller than `1e-4*10.0/3.5`	Period/200

QPSS Options

Commonly Used Options

There are only a small number of options that are commonly used.

© 1994-2020 All Rights Reserved.

stabcycles

If you are having convergence problems, increasing stabcycles usually helps. Increase to about 10. If this does not work, look at the `conv_norm` number at the end of each stabilization cycle, and increase stabcycles until the `conv_norm` does not decrease much with more cycles.

© 1994-2020 All Rights Reserved.

maxperiods

Watch the *conv_norm* number that is reported at the end of each Newton iteration. If this number is generally trending down, then more newton iterations may allow convergence. Start with about 50.

Uncommonly Used QPSS Options

The following options are in the *Convergence* tab.

ic

Initial conditions can be specified graphically by selecting *Simulation* - *Convergence Aids* - *Initial Condition* in ADE Explorer. Initial conditions can also be specified from a file using the *readic* property. Capacitors and inductors have the initial condition properties in the property list for the component. For capacitors, this is an initial voltage across the capacitor, and for inductors, it is an initial current in the inductor. The default is to observe all the initial conditions in the DC analysis that is used as the time-zero timepoint. The initial conditions force a voltage or current to be present in the time-zero solution. The initial conditions are

© 1994-2020 All Rights Reserved.

released for the rest of the transient simulation in the tstab interval. The *ic* option controls the initial conditions that should be observed in the time-zero timepoint. *all* is the default. *dev* means that only the initial conditions on capacitors and inductors are observed. *node* means that only the initial conditions on a node are observed. *dc* means that no initial conditions are observed.

skipdc

In some cases, the time-zero timepoint DC analysis does not converge. Instead of stopping the simulation, *skipdc* allows the tstab simulation to continue using an assumed solution for the time-zero timepoint in the tstab interval. The default is *no* and a DC analysis is run to get the initial timepoint. *yes* means skip the DC solution, and proceed directly to the tstab simulation. All the nodes with initial conditions specified start at the initial condition value. Nodes with batteries start at the battery voltage. Nodes with no initial conditions start at zero volts. For `skipdc=yes`, the signal sources start as specified immediately in the tstab simulation. *sigrampup* uses the same assumptions for the starting voltages as *yes* does, but the start time is set to negative one tenth of the stop time for the tstab interval. At this time, the signal source time-varying part starts at zero and linearly ramps up to the full value at time = zero. After time = zero, the sources have the full amplitude time-varying part.

readic

This specifies an ascii file that contains two columns to be read as initial conditions. The first column is the node name. The second column is the voltage value. If the entry does not start with / (slash), the file is located in the netlist directory. To find the netlist directory, select *Setup- Simulator/Directory/Host* in ADE Explorer. Look in the *Project Directory* field for the location of the simulation directory. Navigate to that directory and then to the `<Circuit Name>/spectre/<schematic or config>/netlist` directory. You can also click (...) and browse to the directory.

maxiters

The transient analysis for tstab and shooting interval iterate to a solution at each timepoint. *maxiters* specifies the maximum number of iterations before the timestep is cut for another try at convergence. In some cases, model parameters can cause discontinuities in the device current or capacitance. If this occurs, the change in the circuit condition when the discontinuity is crossed can be large enough to require more than the five iterations that are allowed by default. Specifying *maxiters* to the 40 to 100 range usually allows the simulator to converge in spite of the discontinuity.

The following options are in the Accuracy tab.

© 1994-2020 All Rights Reserved.

method

This option controls the integration method for tstab and the shooting windows.The default value is *gear2only*, which is recommended for all qpss simulations.

relref

In the order of most to least accurate, the *relref* settings are *pointlocal*, *alllocal*, *sigglobal*, and *allglobal*. The default for the *moderate* and *conservative* accuracy selections is *alllocal* which is preferred. Liberal is not suggested.

relref is used in the transient analysis in the tstab interval for both shooting and harmonic balance and in the shooting window for shooting. In some cases, the timestep can collapse to near zero when the default is used for relref. The symptom is that in the Spectre output window, no progress is made for many reporting intervals. By default, every 10 seconds, Spectre will show progress in the tstab and shooting intervals. For example, for every update in the Spectre output window, the percent complete might stay at 15%. The solution to this is to set *relref* to *sigglobal*, which is slightly less accurate. In the Fourier transform, the noise floor usually degrades by 3 to 6 dB. Another possibility is to try the *Iteminstep* option as described in the *AdditionalParams* option of the *Misc* tab.

Refer to Relref on page 664 for a detailed discussion on *relref*.

Iteratio

Iteratio is a multiplier for the allowable numerical integration error in the tstab interval and the shooting windows. The default is 3.5. *Iteratio* cannot be set smaller than 1.0 and is normally between 3.5 and 100. In some cases, the timestep collapses to near zero In this case, there might be a model discontinuity. To test this, disable numerical integration timestep control by setting *Iteratio* to 1e9. Note that this is an extreme measure not to be used under normal simulations. If the timestep still collapses, there is a discontinuity in one of the models in the circuit. This could be a device model or a Verilog-A model. If it is necessary to set *Iteratio* very large to get the simulation to complete, you must also set *maxstep* small enough to preserve the accuracy of the simulation.

itres

Use the default for this option.

Itres controls the precision (number of digits solved for) in the first Newton iteration. The default is 1e-4 which causes four digits of resolution in the first Newton iteration. Subsequent iterations are solved with more precision.

© 1994-2020 All Rights Reserved.

resgmrescycle

Leave this option at the default.

inexactnewton

In the simulation time between iterations, an estimate is made for the starting point for the next iteration. *inexactNewton* causes an inexact solution for the matrix that calculates the next starting point to be calculated. In some cases, solving the matrix exactly on the first iteration can cause the iterative solver to need many iterations to achieve convergence. Setting i*nexactNewton* causes a different series of iterations, which may speed up the simulation, or alternatively, allow the simulation to converge. The inexactness is reduced as the iterations progress. Try this option when you have convergence difficulties.

The following options are in the *Output* tab.

annotate

This option controls the level of detail in the output log. The default is *status*. No detail is provided when you select *no*, while more details are provided as you move towards right with steps option providing maximum detail.

save

The default is *allpub*. This saves all the public voltages at all levels of the hierarchy in the schematic. It excludes the internal nodes of the device models. *all* adds the internal nodes of all the devices. *lvl* saves all the nodes including the internal nodes of the devices through the level of hierarchy set in the *nestlvl* option. *lvlpub* is like *lvl*, but it does not save the internal nodes. *selected* saves only the nodes that are specifically saved.

nestlvl

If *save* is set to *lvl* or *lvlpub*, this controls the maximum level of hierarchy to be saved. If *nestlvl* is *1*, only the top level is saved. If *nestlvl* is 2, the top level and the next level down are saved. The value for *nestlvl* must be an integer.

compression

Normally, this should not be used for RF simulation. It is not digital compression. This option applies only in the tstab interval of the simulation. For RF simulation, where the input is

© 1994-2020 All Rights Reserved.

sinusoidal, the size of the results file will normally double. It is useful only for circuits that are predominantly square wave.

oppoint

Normally, this should be left at the default of *rawfile*. It controls where the time-zero operating point solution should be saved to.

skipstart

Normally, this is not used for RF simulation unless there is a very long tstab. It is a way of reducing the amount of data in the output file from the tstab interval by not writing many of the timepoints to the output file. *skipstart* controls the simulation time where the skipping is to start. The default is 0 (zero).

If you want to save all the data for the first 5% of tstab, set skipstart to 0.05 * Tstab time. This will save all the data in the first part of the startup interval in tstab, and then begin reducing the amount of data sent to the output file.

skipstop

Normally, this is not used for RF simulation unless there is a very long tstab. This specifies the simulation time in the tstab interval where the skipping is to stop. The default is the stop time in the tstab interval.

If you want to save all the data for the last 5% of tstab, set *skipstart* to 0.95 * Tstab time. This will save all the data in the last part of the startup interval in tstab. This allows full resolution at the end of the startup interval.

skipcount

If *skipstart* and *skipstop* are set, either *skipcount* or *strobeperiod* should be set, but not both. *skipcount* saves one out of every skipcount points to the output data file. A *skipcount* of 3 saves one, then skips two points. Skipcount of 10 saves one in 10 timepoints to the output file. The default is 1, which saves every point.

© 1994-2020 All Rights Reserved.

strobeperiod

If *skipstart* and *skipstop* are set, either *skipcount* or *strobeperiod* should be set, but not both. *strobeperiod* forces the simulation datapoints between *skipstart* and *skipstop* at the interval of *strobeperiod*.

strobedelay

strobedelay specifies a time after the skipstart point for the beginning of the strobing. *strobedelay* must be between 0 (zero) and the time set in *strobeperiod*.

The following options are in the *Reuse* tab.

write

write specifies a filename in which to write the DC solution used as the first timepoint. If the name does not start with slash (/), the file will be written in the *netlist* directory. To determine the netlist directory location, select *Setup - Simulator/Directory/Host* in ADE Explorer. The *Project Directory* field lists the location of the simulation directory. From this directory, the netlist directory is in `<Circuit Name>`/`<simulator name>`/*<schematic or config>/netlist*.

writefinal

writefinal writes the final qpss timepoint from the shooting window (not the final timepoint of the tstab interval) to the filename specified. See the description for *write* for the location.

swapfile

During the shooting intervals, the solution matrices need to be saved in order to calculate the settled voltages of every node for the next iteration. The solution matrices are the internal solution for all the simultaneous equations for the circuit. The solution vector is also saved, and these are the solution voltages and currents. Normally, this data is kept in memory which makes the Spectre process get larger, starting with the first Newton iteration. On 32-bit systems, a maximum process-size of slightly less than 4GB limits the size of the circuit that can be simulated. Even on 64-bit systems, there is a limit set by the amount of memory that is installed on the system that is running Spectre. If the process gets larger than the installed memory, then the process starts swapping to the disk.

Swapping that is done by the operating system is inefficient because there is no logical relationship for the swapped memory pages. Setting swapfile to a name causes Spectre to

© 1994-2020 All Rights Reserved.

write all the solution matrices to the disk in sequential cylinders so that the read time from the disk is faster than using swapping by the operating system. Disk accesses are inherently much slower than memory accesses, however, using swapfile is considerably faster than swapping in the OS. Using swapfile reduces the size of the Spectre process so that it fits in the memory installed in the machine, or allows larger circuits to be simulated. There is a large amount of data that typically is written to the disk. Because of this, the best choice is to write the data in a disk that is local to the machine. If the data is written over a high-latency network, it might actually be better to use the local swap space. Only a trial can determine which is better.

writeqpss

This option specifies that the full internal state of the qpss analysis be written out to the file specified in the option.

readqpss

This option specifies the file to be read in as a starting point for the next qpss analysis. If nothing is changed in the circuit, then only one Newton iteration is performed and small-signal analyses can be run with a much faster time for the qpss analysis. Small changes can be made to the circuit or to the analysis options as long as the topology of the circuit stays the same. Changes introduce a discontinuity at the beginning of the qpss analysis, which might be large enough to cause the qpss analysis to not converge.

saveclock

saveclock saves the state of the simulation in the tstab interval at the time interval in seconds specified by *saveclock*. When the clock time has passed in the tstab interval, the file specified by the option *savefile* is created. When subsequent clock time intervals have passed, the file is overwritten. Use only one of the save options at a time.

saveperiod

saveperiod saves the state of the simulation in the tstab interval at the simulation time interval in seconds specified by *saveperiod.* When the simulation time has passed in the tstab interval, the file specified by the option savefile is created. When subsequent simulation time intervals have passed, the file is overwritten. Use only one of the save options at a time.

© 1994-2020 All Rights Reserved.

savetime

savetime is a list of times in the tstab interval where the state of the simulation is written out. The list should be specified with spaces between the entries. The information is written out with the filename specified in the *savefile* option with extensions for the time after that. Use only one of the save options at a time.

savefile

This is the file name to write out the tstab state information to. If you do not specify a filename that starts with the slash (/) character, the file is stored in the *netlist* directory. To locate the *netlist* directory, first select *Setup - Simulator/Directory/Host* menu in ADE Explorer, and read the path to the project directory. This is the path to the *simulation* directory. In the simulation directory, navigate to the *<`circuit_name`>/spectre/<schematic or config>/netlist* directory.

Do not specify a relative path name like *./save_file/run1*.

recover

recover specifies the file name to recover the tstab simulation from. If *saveperiod* or *saveclock* are used to make the savefile, then just the same name as specified in *savefile* is used. If *savetimes* has a list of times specified, then several files are created at the times

© 1994-2020 All Rights Reserved.

specified in the list. Click (...) and browse to the netlist directory where the savefiles are shown.

Select one from the list, and click *Open*. This adds the full path to the *recover* option, as shown below. Also note that in the example below, the *savefile* option is still set, and times after the restart times are specified. This is specifically allowed. As long as tstab is 200n or larger, the files at 175n and 200n will be added and can be reused later.

© 1994-2020
All Rights Reserved.

The following options are in the *Misc* tab.

step

Leave this option at the default. All the nodes are checked for curvature in the local truncation error (Numerical integration error) check. If that node has a capacitor or inductor connected to it, the timestep should be controlled to reduce the numerical integration error. If the node does not have a capacitor or inductor connected to it and it has a lot of curvature, the timestep can only be reduced to the value specified in the *step* option. The default is the simulation interval divided by 1000.

maxstep

In qpss shooting, *maxstep* is observed in the tstab interval and in each of the shooting windows. Normally, *maxstep* should be left at the default and either setting *errpreset* or changing *reltol* is used to control the accuracy. In some cases, extreme accuracy is needed, and this is accomplished by setting a small maxstep value. A small constant timestep produces maximum accuracy at the cost of runtime and memory consumption.

readns

This specifies an ascii file with the same format as an ic file that is used as nodesets for the time-zero DC solution. Nodesets do not force a voltage to be held for the time-zero solution. Instead, they are a way of speeding up the time-zero calculation. As a suggestion, set the *write* option and the *readns* option to the same filename. The *write* option writes the time-zero solution to a file. When this is used as a starting point, many fewer iterations are needed for the time-zero point to converge.

cmin

cmin can be used to improve convergence in the tstab and shooting intervals. If a value is set for *cmin*, a capacitor with this value is added to every node with the other terminal of the capacitor connected to the global ground node. If a 10f to 50f capacitor is added, this prevents instantaneous changes from occurring from timepoint to timepoint, thus improving the convergence at the cost of adding non-physical capacitors to the circuit. An example with 10 femtoFarads is shown below. Note that if 10 is entered, a 10 Farad capacitor is added from every node to ground. Remember the multiplier in the entry. An example is shown below.

© 1994-2020 All Rights Reserved.

tstart

Normally, this is not used. *tstart* specifies the starting time for the tstab interval and defaults to 0 (zero). A negative, 0, or positive *tstart* is allowed.

additionalParams

This is a field for `<option_name>=value` statements. Multiple `<option_name>=value` statements are allowed with a space between them. Generally, this is used to unlock the beta features, however, there is one case where this field might be useful for an option that is not in the GUI.

In some cases, either in the tstab interval or in one or more of the shooting windows, the timestep collapses to near zero. The symptom is that the percent complete stays at the same value for many reporting intervals. In this case, there might be a discontinuity in one of the models.

In transient simulation, two things can reduce the timestep; having too much numerical integration error or taking too many iterations at a single timestep can cause the timestep to be cut. Spectre does not report what is causing the timestep to become small. One way to eliminate the possibility of numerical integration error (which is called local truncation error in the simulator) as the cause of reducing the timestep is to set *lteratio* to 1e9. The disadvantage of this is that it disables the normal method of timestep control, therefore, *maxstep* must be set to maintain accuracy.

Another way to accomplish this is to set *lteminstep* to a value of about the stop time divided by 1e5. This allows the normal method of timestep control, but if for some reason the numerical integration error wants to cut the timestep too small, the error is ignored. *lteminstep* is the way to specify this. To do this in ADE Explorer, type `lteminstep=<Stop_Time>/1e5`. If you use `lteratio=1e9`, or if you set *lteminstep* and you still have very small timesteps, try increasing *maxiters* in the *Convergence* tab to between 40 and 100. When *lteratio* or *lteminstep* is set and you still have timestep problems, there is likely a discontinuity in either a device model or in a Verilog-A model. An example is shown below.

© 1994-2020 All Rights Reserved.

Example

Consider a sample and hold circuit below.

© 1994-2020
All Rights Reserved.

1. The input signal properties are shown below.

2. The input frequency and amplitude terms are set to variable names so that the amplitude and frequency can be easily set in ADE Explorer. Also, note that the sinusoid is centered at 1.55 volts. This is because the simple circuit above runs between 0 volts and 3.3 volts.

© 1994-2020
All Rights Reserved.

3. Next, the variable values are set in ADE Explorer. The sample clock is 100MHz, and the signal is at 1MHz.

© 1994-2020 All Rights Reserved.

4. Now, the qpss analysis is set up, as shown below.

5. All the sources in the circuit are listed in the table at the top. The sample clock is set to large for two reasons. First, it causes the most distortion in the circuit. Second, it is the nonsinusoidal source. The input signal is set to moderate. 15 sample clock harmonics are

© 1994-2020 All Rights Reserved.

set, and 5 harmonics for the input signal are set. Conservative is used because the distortion is likely to be small at low input amplitude.

6. Now, run the simulation. When the run is complete, select *Results - Direct Plot - Main Form* in the ADE Explorer window. The *Direct Plot Form* is displayed.

© 1994-2020
All Rights Reserved.

In the *Direct Plot Form*:

7. Select *qpss*. In this case, the voltage spectrum is plotted. Peak is used because it allows a direct comparison of input amplitude in volts peak to the output amplitude in volts peak. *Add To Outputs* is selected so that an expression gets added under the Maestro tab of the ADE Explorer window.

8. The result is displayed in the waveform window.

© 1994-2020
All Rights Reserved.

9. Since the output at low frequency is desired, the X Axis is scaled to show just the output terms and the harmonics. To place a marker, move your mouse cursor near the frequency point you want, and type m.

10. The output is near 50mv, as set in the input source.

Add an ADE expression to calculate the amplitude of the output at the input frequency.

When you select the *Add to Outputs* check box in the Direct Plot Form and plot the results, the expression is added in the *Details* column, under the *Maestro* tab of ADE Explorer, as shown below.

© 1994-2020 All Rights Reserved.

© 1994-2020 All Rights Reserved.

1. Click twice on the expression and then click the *Open calculator* button, as shown below.

Note: Starting with the IC615 ISR12 release and later, the RF expressions are much simpler than they were in the past.

© 1994-2020 All Rights Reserved.

The calculator starts with the expression in the buffer.

© 1994-2020
All Rights Reserved.

2. In the calculator, select the *mag* function to convert the complex voltage to a scalar voltage. The assumption here is that the load is resistive, and any voltage regardless of phase will create power.

© 1994-2020 All Rights Reserved.

3. In the calculator, select the *value* function to get the value at 1MHz and click *OK*.

4. Select the *Send buffer expression to ADE Outputs* button, as shown below.

© 1994-2020 All Rights Reserved.

Alternatively, you can click *Tools - Send Buffer to ADE Outputs*.

© 1994-2020 All Rights Reserved.

This adds the expression under the Maestro tab of ADE Explorer, as shown below.

© 1994-2020 All Rights Reserved.

5. Specify a name for the expression. Here, it shows HARM1.

In a similar manner, add expressions for all 5 harmonics of the output

© 1994-2020 All Rights Reserved.

6. To verify that the expressions work, click the *plot Outputs* icon located at the right of the ADE explorer window.

Now build a calculator expression to calculate THD.

THD in percent is defined as: 100* square root (total power in the harmonics divided by the power in the fundamental).

First, build an expression to calculate the power in the harmonics.

1. Open the HARM2 expression in the Calculator window by clicking the *Open calculator* button.

© 1994-2020 All Rights Reserved.

2. This adds the expression into the calculator.

© 1994-2020 All Rights Reserved.

3. In the calculator, select the *x**2* function. (Square the expression) This converts voltage to power.

© 1994-2020
All Rights Reserved.

4. Press the <*Enter*> key three times. You may need to click *View* - *Show Stack* to view the stack. This is the beginning of adding the power in the higher harmonics.

Note: The number of times you press the <*Enter*> key will depend on the number of harmonics that are specified in the qpss analysis. Here, there are five total harmonics, the fundamental and four higher harmonics. The total number of higher harmonics is four. To get four total expressions, you press the <*Enter*> key three times.

© 1994-2020 All Rights Reserved.

5. Click + (the plus sign) three times. Now change the frequencies in the buffer so you have 2M + 3M +4M +5M. This adds the power in the higher harmonics.

© 1994-2020 All Rights Reserved.

6. Verify that the expression is legal by clicking *Evaluate the buffer and display the results in a table*. This is an insurance step that is always recommended when you build calculator expressions. The icon for the table output is shown in the red box below.

7. The value is shown below.

© 1994-2020
All Rights Reserved.

8. Add parentheses around the entire expression using the mouse and keyboard.

© 1994-2020 All Rights Reserved.

9. Press the <*Enter*> key on your keyboard. The expression is entered into the stack.

© 1994-2020 All Rights Reserved.

10. Open the HARM1 expression in the calculator.

© 1994-2020 All Rights Reserved.

11. Click the X**2 special function. This calculates the power in the first harmonic.

© 1994-2020 All Rights Reserved.

12. Select the / (divide by) key. This divides the power in the harmonics by the power in the fundamental.

© 1994-2020 All Rights Reserved.

13. Verify that the expression is legal by clicking the *Evaluate the buffer and display the results in a table* button in the calculator.

14. The result is shown below.

© 1994-2020
All Rights Reserved.

15. Select the sqrt (square root) special function.

© 1994-2020 All Rights Reserved.

16. Verify that the expression is legal by clicking the *Evaluate the buffer and display the results in a table* button in the calculator.

17. The value is shown below.

© 1994-2020
All Rights Reserved.

18. Click at the beginning of the expression, and type 100*.

© 1994-2020 All Rights Reserved.

19. Verify that the expression is legal by clicking the *Evaluate the buffer and display the results in a table* button in the calculator.

20. The value is shown below.

21. In the Calculator window, choose *Tools - Send Buffer to ADE Outputs*.

The expression is added under the *Maestro* tab of the ADE Explorer window.

© 1994-2020
All Rights Reserved.

22. Type *THD* in the *Name* field.

.

© 1994-2020 All Rights Reserved.

Now sweep the input level, and run the simulation.

1. In the ADE window, double-click the qpss line under the Analysis section in the Setup pane. The qpss *Choosing Analyses* Form is displayed.

2. Select the check box to the right of the Sweep drop-down list.

© 1994-2020 All Rights Reserved.

3. Select the appropriate design variable that sets the input amplitude.

4. Specify an appropriate range and step size for your application. In this example, 50mv to 1.1v is the sweep range, with linear spacing. Click OK, and run the analysis. When the simulation completes, the amplitude of the first five harmonics, and the THD plot, as shown below.

5. Markers are placed at the input amplitude of 1 volt peak. At that level, the output is 970mV peak, and the THD is 2.99%.

Add an expression to measure the clock feedthrough

1. Open the HARM4 expression in the calculator by click the Open calculator button.

© 1994-2020
All Rights Reserved.

2. This adds the expression for the 5MHz term into the calculator. In the calculator, change 5M to 100M.

© 1994-2020 All Rights Reserved.

3. In the Calculator window, choose *Tools - Send Buffer to ADE Outputs*. The expression is added under the *Maestro* tab of the ADE Explorer window.

4. Name the expression *Sample_Feedthrough*.

© 1994-2020 All Rights Reserved.

5. Deselect all the check boxes under in the *Plot* column for all the expressions except for *Sample_Feedthrough*.

6. Click the *Plot Outputs* icon in ADE Explorer.

   This plots the clock feedthrough.

   Remember that this is a peak measurement.

© 1994-2020
All Rights Reserved.

Now plot the time-domain waveforms

1. In the waveform tool, click *File - Close All Windows*.

© 1994-2020
All Rights Reserved.

2. In ADE Explorer, click *Results - Direct Plot - Main Form*. The *Direct Plot Form* is displayed.

3. In the *Direct Plot Form,* select *ifft* from the *Sweep* section

4. Select *Complete ifft*.

5. Select *100m*.

6. Click *Replot*.

7. Select *300m*.

© 1994-2020
All Rights Reserved.

8. Click *Replot*.
9. Select *500m*.
10. Click *Replot.*
11. Select *700m*.
12. Click *Replot*.
13. Select *900m*.
14. Click *Replot*.
15. Select *1.1*.
16. Click *Replot*.

Although the distortion is not visible until the input amplitude becomes 1.1 volts, there is distortion in the output waveform at the lower levels.

© 1994-2020
All Rights Reserved.

Quasi-Periodic AC Analysis (QPAC)

Refer to the pac analysis section in Chapter 4 for details about the pac analysis.

Qpac is just like pac, except it runs after a qpss analysis instead of after a pss analysis. Qpss captures the harmonics of the system, and qpac applies a single input frequency at a time, and calculates the output mixing products when the qpac input mixes with the harmonics of qpss. Conceptually, the only difference from pss-pac is that qpss has more signals and more harmonics in it to mix with, so there are more output mixing terms.

Example

Consider a diode mixer below.

© 1994-2020
All Rights Reserved.

The input port has the frequency and amplitude set using variable names. This is done in order to allow the frequency and amplitude changes in ADE Explorer without the need for changing the schematic.

© 1994-2020
All Rights Reserved.

PAC Magnitude is set to *1 V* in the port. *PAC Magnitude* is used by pac and qpac to define the amplitude of the input signal for both analyses. This allows convenient measurement of conversion gain. Since the input is set to 0 dB, the conversion gain can be read directly just by probing the desired net.

In a similar manner, the LO signal uses variable names. These variables are set in the ADE Explorer window.

© 1994-2020 All Rights Reserved.

Next, the qpss analysis needs to be set up. For more information, see the qpss section at the beginning of this chapter.

© 1994-2020
All Rights Reserved.

Next, the qpac analysis needs to be set up.

1. The input is from 850MHz to 950MHz, with a linear sweep.
2. Because 10 harmonics were set in the qpss analysis for the LO tone, 10 is set for *Maximum clock order*. These numbers should agree.
3. Now run the analysis.

© 1994-2020 All Rights Reserved.

4. When the simulation completes, select *Results - Direct Plot - Main Form* in the ADE Explorer window. The *Direct Plot Form* is displayed.

5. Select *qpac* from the Analysis section.

6. In many cases, *dB20* is useful.

© 1994-2020
All Rights Reserved.

7. The output net in the schematic is selected, The conversion gain terms are plotted, as shown below.

Because there are a large number of harmonics in the circuit, there are a large number of mixing products that are produced. In the view below, zooming is applied to show just the

© 1994-2020
All Rights Reserved.

main mixing products. Markers have been placed for the center frequency of the sum and difference frequency mixing terms.

As is common for diode mixers, the insertion loss is about 6 dB.

Qpac Concepts

Generally, when qpac is needed, harmonic balance is appropriate for the simulation engine. In this case, the hb and hbac analyses are recommended because the hb and hb analysis forms always have the latest features.

Qpac applies a small-signal input to the system that is simulated with the qpss analysis. Qpss solves for the harmonics of the system with two or more inputs to the circuit. Qpac calculates the small-signal outputs that are created by mixing with all the harmonics that are in the qpss solution.

One application of qpss-qpac is measuring conversion gain of an LNA-Mixer circuit when a large amplitude blocking signal is present. In this case, the LO and the blocker are applied in qpss, and the small-signal conversion gain is measured using qpac. Note that for this application, harmonic balance is likely to run faster than shooting, and as a strong recommendation, use the hb and hbac analysis for this application.

© 1994-2020 All Rights Reserved.

Unlike hbac, qpac has only one mode, which measures the small-signal mixing products. Modulated, sampled, rapid IP3 and IP2, and the distortion summaries from hbac are not available in qpac.

Down Conversion: Input Near the First harmonic of the LO

Qpac is for measuring the conversion gain that is the average conversion gain with all the signals applied to the circuit (usually the LO or the sample clock and a blocking signal) applied to the circuit. The mixing products that are produced are called sidebands. The sideband number has a value for each signal that is applied to the circuit, and is the harmonic number that is being mixed with to provide the output. A positive number means that the output mixing product is higher in frequency than the input frequency specified in the *Choosing Analyses* form. A negative number means that the output is lower in frequency than the input frequency. If there are two inputs to the circuit in the qpss analysis, there will be two numbers for each qpac sideband. In the example below, there is an LO at 1.8GHz and a blocker at 2GHz. The RF small-signal input is at 1.85GHz. A waveform plot is shown below. The red spectrum is the spectrum calculated by the qpss analysis, and the small dots are the qpac result. The amplitude of the qpac output is larger than the qpss output because the qpac outputs assume a 1 volt input, and the qpss has much smaller input levels. It will explain further.

© 1994-2020
All Rights Reserved.

All the sideband numbers are plotted on the left of the waveform tool. When 0 0 is selected, the output frequency is the input frequency that does not mix with either input frequency. Thus, the 0 0 sideband is at 1.85GHz (the input frequency) as shown below.

© 1994-2020
All Rights Reserved.

The primary output mixing product is the signal that mixes down with the LO, and does not mix with the blocker. It mixes down with the first harmonic of the LO, and does not mix with the RF blocker, thus the sideband number is -1 0.

1.85GHz (the input frequency) minus 1*LO frequency (1.8GHz) is 50MHz.

© 1994-2020
All Rights Reserved.

1.85GHz can also mix down with the blocker frequency. In this case, the signal mixes down with the first harmonic of the blocker (2GHz) and not with the LO.

The sideband number here is 0 -1. 1.85GHz -1*2GHz = -150MHz. By default, this is mirrored up to the positive frequency domain in the qpac analysis.

© 1994-2020 All Rights Reserved.

Next, the frequency can mix up with the LO. This is the 1 0 sideband.

To calculate the frequency, the input is 1.85GHz + 1* 1.8GHz = 3.65GHz.

© 1994-2020
All Rights Reserved.

The Easy Way to Calculate Sideband Numbers and Frequencies

The easy way to calculate sideband numbers and frequencies is to use the *Direct Plot Form*. The *Direct Plot Form* displays the sideband numbers and the frequency ranges when you select *sideband*, as shown below.

In this example, the LO is at 1.8GHz, the RF Blocker is at 2GHz, and the input sweep range is 1850MHz to 1851MHz.

© 1994-2020
All Rights Reserved.

Avoid Wide Sweep Ranges

Wide sweep ranges cause very confusing results. This should be avoided. An example is shown below.

The marker is placed at 1.7GHz. The value from qpac at the marker value is shown in the legend area on the left side of the waveform display. The different traces result from mixing with different harmonics in the qpss analysis.

Relative Frequency Sweep

The qpac analysis is like AC, where the input frequency is swept. The input frequency range is specified in the qpac *Choosing Analyses* form.

In addition to the frequency, there is a selection called *Sweep Type*. The choices are *relative* and *absolute*. When *absolute* is selected, the frequency range is used directly with no frequency conversion. When *relative* is selected, a *Relative Harmonic* field appears. The frequency sweep is shifted by the specified frequency of the harmonic numbers in the qpss analysis.

© 1994-2020 All Rights Reserved.

For example, assume that the qpss has a 1GHz large-signal input, and a 900MHz *Moderate* input. You want a log sweep above the first harmonic of the 1GHz signal in the qpss analysis. In this case, you need to select *relative* sweep, and specify 1 0 in the *Relative Harmonic* field. The first number is the frequency shift in multiples of the LO frequency, and the second number is the frequency shift in multiples of the RF input frequency. Next, type in 1K to 100M for the frequency range with a log sweep and 3 to 5 points per decade.

The sweep will start at 1G + 1K, and sweep to 1G + 100M using log spacing from 1K to 100M. This is illustrated below.

To use a log sweep below the first harmonic of the 1GHz input, use the same frequency range, and type in -1 0 as the relative harmonic. This is illustrated below.

© 1994-2020
All Rights Reserved.

Maximum Clock Order

In this case, clock is the terminology for the large signal in the qpss analysis. Sidebands (the output mixing products calculated by qpac) are calculated through the highest harmonic of the large signal in the qpss analysis. This is usually the LO signal.

Conceptually, an infinite number of mixing products are produced when a single input is applied to a nonlinear system. From the practical point of view, usually only a small number of mixing products need to be measured.

Consider an example. Assume that a 1GHz large signal and a 900MHz moderate signal is applied in qpss. 10 harmonics are set for the large signal at 1GHz, and three harmonics are specified for the moderate signal. Assume there is no need to measure the mixing products above the third harmonic of the LO because there is no spec at frequencies above that. In that case, you could set *Maximum clock order* to three, and only the mixing products that are produced by mixing with the first three harmonics of the LO are calculated in the qpac analysis.

Sideband is the name of the qpac output mixing product. The sideband number is the qpss harmonic numbers that is being mixed with. If the sideband number is negative, the output frequency is mixed down in frequency from that input. If the sideband is positive, the output is mixed up in frequency. A sideband contains the same number of numbers as there are tones in the circuit. If there are 2 tones in the circuit, each sideband has two numbers, like -1 0. If there are three inputs, each sideband has three numbers, like -1 0 0.

Setting Harmonics in QPSS to get Accurate QPAC Results

The qpss analysis, when shooting is selected, has a minimum of 200 timepoints in each slice, so the large signal in qpss inherently has frequency domain content through the 100th harmonic of the large signal input. Only the harmonics that are specified in the moderate tone are included in the solution. The impact of this is that the qpac mixing products that are produced from mixing with the large signal are almost always correct, but the mixing with the moderate signal might or might not be accurate, depending on the number of harmonics that are specified for the moderate signal, and the number of harmonics actually produced by the system. Like harmonic balance, you need to run the qpss and qpac with a number of harmonics specified for the moderate tone(s), and make the measurement in qpac. Then increase the number of moderate harmonics by about 50%, and run again. Then note the measurement. If the result changed significantly, then you need more moderate harmonics. If the result did not change, you might be able to reduce the number of moderate harmonics. Use the smallest number of moderate harmonics that produce a stable qpac result.

© 1994-2020 All Rights Reserved.

ADE Implementation

In this section, just the qpac analysis *Choosing Analyses* form and the options for qpac are shown. An examples section follows with all the steps needed for a simulation.

To open the *Choosing Analyses* form, select *Analysis* - *Choose* in ADE Explorer, or click he *Choosing Analyses* icon ( )on the right of the ADE Explorer window.

The qpac *Choosing Analyses* form appears.

The frequency range is the input frequency range. It is usually better to select Linear or Log spacing, and define your own sweep because auto is likely to take more points (and time) than you need. *Maximum clock order* specifies the highest qpss large signal harmonic to calculate mixing products for.

© 1994-2020 All Rights Reserved.

If you need a log sweep around a harmonic, set the Sweeptype to *Relative*.

The number of numbers in the relative harmonic must be equal to the number of qpss tones. If there are 2 qpss tones, there should be two numbers in the relative harmonic. The first number is the harmonic number and direction that the qpac tone mixes with for the large tone in qpss. The second number specifies the harmonic number of the moderate tone that the input frequency mixes with. Generally, the first tone is the LO tone, and the second tone is the RF blocker. Generally, we design to mix with he LO, and not mix with the blocker, thus the first number is usually 1 or -1 because generally we mix with the first harmonic of the LO, and generally the second number is zero because we don't deliberately mix with the RF blocker.

QPAC Options

None of the qpac options are commonly used.

© 1994-2020 All Rights Reserved.

tolerance

Leave this option at the default value.

Qpac uses an iterative solver to calculate the output amplitudes. Any iterative solver needs an error tolerance to specify when to stop iterating because the solution is accurate enough, and the tolerance option specifies that accuracy for the qpac analysis. For shooting, the default tolerance is 1e-9. When harmonic balance is the qpss engine, the default is 1e-6.

gear_order

Do not change this option.

solver

The default solver is the turbo solver.

When hb is used as the engine in qpss, leave this option at the default.

When shooting is used, sometimes when the qpac input frequency is very close to the frequency of one of the harmonics in the qpss, warning messages will appear in the qpac output warning that the accuracy might not be good enough. If you see these messages, select the std solver, which has better ability to handle frequencies that are very close to a harmonic in the qpss solution, but which takes longer to run than the turbo solver.

resgmrescycle

Leave this option at the default. For the resgmres linear solver, there are several different options.

hbprecond_solver

This option is only available only when harmonic balance is selected in qpss.

The basic solver is the only solver available in standard Spectre. *autoset* is the default solver when APS is used. This solver is faster, but occasionally stagnates. When stagnation is detected, APS automatically switches to the *basic* solver, and prints a message in the Spectre output window.

© 1994-2020 All Rights Reserved.

annotate

Selections to the left produce less qpac output in the Spectre output window. No detail is provided when you select no, while more details are provided as you move towards right with steps option providing maximum detail.

freqaxis

freqaxis specifies whether you want to see the negative frequency axis or not.

The analysis calculates the output frequencies and amplitudes, so *out* and *absout* are reasonable choices. *out* displays the negative frequency axis. *absout* (absolute value of the output) displays positive output frequencies. This is the default.

If you select *in*, all the outputs at the different frequencies are plotted on the same input frequency range scale. This is not recommended.

save

The default is *allpub*. This saves all the public voltages at all levels of the hierarchy in the schematic. It excludes the internal nodes of the device models. *all* adds the internal nodes of all the devices. *lvl* saves all the nodes including the internal nodes of the devices through the level of hierarchy set in the *nestlvl* option. *lvlpub* is like *lvl*, but it does not save the internal nodes. *selected* saves only the nodes that are specifically saved. In ADE Explorer, this is accomplished using the *Outputs - To Be Saved - Select On Schematic* and then selecting the nodes and terminals specifically in the schematic. At the netlist level, this is accomplished by using a save statement with a list of names to be saved.

nestlvl

If *save* is set to *lvl* or *lvlpub*, this controls the maximum level of hierarchy to be saved. If *nestlvl* is *1*, only the top level is saved. If *nestlvl* is 2, the top level and the next level down are saved. The value for *nestlvl* can be any integer.

additionalParams

additionalParams is typically used for new features that are being beta tested. It is a field where `keyword=value` pairs are entered.

For more information about the other options, type `spectre -h qpac` at the command prompt in a Unix shell window

© 1994-2020 All Rights Reserved.

Example

Consider a mixer circuit below.

© 1994-2020
All Rights Reserved.

The RF source properties list is shown below.

© 1994-2020
All Rights Reserved.

PAC magnitude controls the amplitude of the qpac input signal. Setting the amplitude to 1 volt allows the conversion gain to be read directly from the output waveform.

The large-signal properties are set to variable names so the frequencies and amplitude can be controlled using variables in ADE Explorer.

In this example, the LO is at 1.9GHz, and the RF input is at 2GHz. The second RF input is disabled for the qpss analysis by setting its frequency to zero.

1. Now select *Analysis* - *Choose*, or click the *Choosing Analyses* icon ( )in the ADE Explorer window.

© 1994-2020 All Rights Reserved.

2. Select qpss. The qpss analysis must be run before a qpac can run. For more information on setting the qpss form, see the qpss section at the beginning of this chapter.

3. If there is a non-sinusoidal source, it must be set to large. This is a requirement even if the non-sinusoidal signal produces only a small amount of distortion. Otherwise, the large signal should be the signal that causes the most distortion in the system.

© 1994-2020 All Rights Reserved.

4. Set harmonics large enough to capture the actual harmonics that are produced by your circuit.

5. Now select *qpac*.

6. This setup causes a log sweep just above the first harmonic of the LO signal. The qpss analysis has 2 tones in it, so the *Relative Harmonic* field has two numbers in it. The first number is the harmonic number of the large signal that is being mixed with, and the second number is the harmonic number of the moderate tone that is being mixed with. 1 0 causes the input frequency to be shifted up by mixing with the first harmonic of the LO signal. This puts the input frequency just above the LO frequency.

7. It is usually better to select a Linear or Log sweep and specify your own frequency points because automatic will likely take more points (and runtime) than is needed.

© 1994-2020 All Rights Reserved.

8. *Maximum clock order* specifies the highest harmonic of the LO tone to calculate outputs for. If you do not need to measure high order mixing products, this can be set fairly small. If you use the harmonic balance engine in qpss, it is suggested that you set *Maximum sideband* to the same value used for the number of LO harmonics in the qpss, or leave the field blank, which does the same thing.

9. Run the simulation. When the simulation completes, select *Results - Direct Plot - Main Form*. The *Direct Plot Form* is displayed.

10. Select *qpss* from the *Analysis* section.

11. Select the output net in the circuit.

12. Click OK.

© 1994-2020
All Rights Reserved.

The spectral plot is shown below.

.

© 1994-2020
All Rights Reserved.

13. In this example, the display has been zoomed in for more detail, as shown below.

© 1994-2020
All Rights Reserved.

14. In the *Direct Plot Form*, select *qpac* from the *Analysis* section.

© 1994-2020 All Rights Reserved.

15. Select *sideband* from the *Sweep* section. This allows the plotting of the individual output mixing products. Select the term you want from the list, and select the desired net in your circuit. The output of all the sidebands in range of the X axis is shown below.

16. The qpac output has a larger amplitude than the qpss output because the qpac input amplitude is set to 1 volt, and the qpss amplitudes are much smaller.

© 1994-2020
All Rights Reserved.

17. To measure conversion gain, go to the desired output frequency, and position a marker as shown below, or use the tracking cursor in the waveform tool.

18. The conversion gain is about 13.5dB.

© 1994-2020
All Rights Reserved.

19. Now sweep the input power of the RF blocker in the qpss analysis. Select the check box next to *Sweep*, and specify a sweep range that is appropriate for your system.

© 1994-2020
All Rights Reserved.

20. Set the qpac analysis to a single frequency, and set the input frequency in the passband.

© 1994-2020
All Rights Reserved.

21. Now run the simulation. When the simulation completes, select *Results - Direct Plot - Main form* in the ADE Explorer window. The *Direct Plot Form* is displayed.

22. Select *variable* from the *Sweep* section, and select the output frequency. Usually, dB20 is appropriate. Now select the output net in the schematic.

© 1994-2020 All Rights Reserved.

23. The Waveform displays. As expected, the conversion gain goes down when the blocker amplitude gets large.

© 1994-2020
All Rights Reserved.

Quasi Periodic Noise Analysis

Overview

The concepts of qpnoise analysis are exactly the same as pnoise analysis, except that instead of running a pss analysis to capture the nonlinearity of the circuit, qpss captures the nonlinearity of the circuit. In qpss, there are multiple inputs which cause many harmonics to be produced by the circuit. All of these frequencies in the circuit can mix with noise frequencies to be converted to the output frequency of the qpnoise analysis. Qpnoise calculates all the frequency translations from all the harmonics in the qpss analysis, and sums the noise power at the output of the circuit.

The normal application of qpnoise is to measure the noise figure degradation of lna-mixer chains with a large amplitude RF blocking signal applied to the circuit.

Note that for this application, harmonic balance is likely to run faster than shooting. If you need harmonic balance, the hb and hbnoise analyses are strongly recommended because these analyses always reflect the latest features that are available in the harmonic balance simulation.

© 1994-2020 All Rights Reserved.

Example

Consider a diode mixer below.

© 1994-2020
All Rights Reserved.

The input port has the frequency and amplitude set using variable names. This is done in order to allow frequency and amplitude changes in ADE Explorer without the need for changing the schematic.

© 1994-2020
All Rights Reserved.

In a similar manner, the LO port uses variable names. These variables are set in the ADE Explorer window.

© 1994-2020
All Rights Reserved.

1. Next, the qpss analysis needs to be set up. For more information, see the qpss section at the beginning of this chapter. In this example, a 1GHz LO signal and a 900MHz blocker are applied.

© 1994-2020
All Rights Reserved.

2. Next, the qpnoise analysis needs to be set up.

© 1994-2020
All Rights Reserved.

3. The output is from 50MHz to 150MHz with a linear sweep. Because 10 harmonics were set in the qpss analysis for the LO tone, 10 is set for *Maximum clock order*. These numbers should agree. If harmonic balance is the engine in qpss, this field can be left blank, which will cause all the harmonics in the qpss analysis to be considered in qpnoise.

4. The output port is set. This subtracts the noise from the output port from the noise figure calculation. The input port is also set. This is required if you want a noise figure calculation.

5. The reference sideband defines the passband frequency for the noise figure calculations. The passband for this case is just under the LO input at 1GHz.

6. Now run the analysis. When the simulation completes, select *Results - Direct Plot - Main Form* in the ADE Explorer window. The *Direct Plot Form* is displayed.

© 1994-2020
All Rights Reserved.

7. The single-sideband noise figure is selected. Click *Plot*.

As expected, the noise figure is very near 6dB, which is the conversion loss for the mixer.

© 1994-2020
All Rights Reserved.

Qpnoise Concepts

Unlike pnoise and hbnoise, qpnoise has only one mode. Qpnoise can only sum the noise power at the output, and refer that noise back to the input. Modulated and sampled functions are not available. If you need those functions, use either pss-pnoise or hb-hbnoise.

Because there are many harmonics created in the qpss analysis, many transfer functions need to be calculated from the different noise frequencies to the output.

Consider an LNA-Mixer with a 2.4GHz LO and an RF signal at 2.45GHz. The spectrum at the output of the circuit is shown below.

If the noise output frequency was 1MHz, then noise at 1MHz on either side of each harmonic in the qpss analysis mixes down to 1MHz. The frequency response of each noise source is determined by the qpss solution. This solution also supplies the nonlinearity to qpnoise so qpnoise can calculate the transfer function from each noise source to the output. The noise power at each noise frequency times the conversion gain gives the noise power at the output. All the different noise outputs caused by each noise frequency mixing down is summed, and then this same process is repeated for each noise source in the circuit.

© 1994-2020 All Rights Reserved.

A zoomed in view near the LO frequency is shown below.

The noise 1MHz on either side of these harmonics mixes with each of these harmonics to the output at 1MHz. These frequencies are shown with the blue arrows above. Note that this shows the spectral content at the output of the circuit only. The amplitudes of these signals are much larger in other parts of the circuit. Noise that is 1MHz above and below all the harmonics mix down to the output at 1MHz. Because the noise frequencies are slightly different for all the mixing products, the actual nose power and the conversion gain is slightly different for each mixing product.

© 1994-2020 All Rights Reserved.

Measuring Noise Figure

The formulas for measuring noise figure are the same as those presented in pnoise.

For:

`No` = total output noise
`Ns` = noise at the output due to the input probe at the passband frequency (the source)
`Nsi` = noise at the output due to the image harmonic at the source
`Nso` = noise at the output due to harmonics other than input at the source
`Nl` = noise at the output due to the output probe (the load)
`IRN` = input referred noise
`G` = gain of the circuit
`F` = noise factor
`NF` = noise figure
`Fdsb` = double sideband noise factor
`NFdsb` = double sideband noise figure
`Fieee` = IEEE single sideband noise factor
`NFieee` = IEEE single sideband noise figure

then,

```
IRN = sqrt(No^2/G^2)
F = (No^2 - Nl^2)/Ns^2
NF = 10*log10(F)
Fdsb = (No^2 - Nl^2)/(Ns^2+Nsi^2)
NFdsb = 10*log10(Fdsb)
Fieee = (No^2 - Nl^2 - Nso^2)/Ns^2
NFieee = 10*log10(Fieee)
```

Note that in order to measure noise figure, the passband frequency needs to be identified. Qpnoise calculates many transfer functions from the input to the output. The passband frequency is set in the Reference sideband parameter. The reference sideband is the sideband number that defines the passband frequency. If there are two inputs in qpss, there will be two numbers in the reference sideband. If there are three inputs, there will be three numbers. The easiest way to define the reference sideband is to use the *Select from List* capability, and then select the design input frequency from the list.

Frequency Sweep

In qpnoise, the fundamental quantity that is calculated is the noise power at the output of the circuit. The Frequency Sweep Range is always the **output** frequency range for qpnoise. The output frequency range is specified in the qpnoise *Choosing Analyses* form.

In addition to the frequency, there is a selection called *Sweep Type*. The choices are *relative* and *absolute*. When *absolute* is selected, the frequency range is used directly with no

© 1994-2020 All Rights Reserved.

frequency conversion. When *relative* is selected, a *Relative Harmonic* field appears. The frequency sweep is shifted by the specified frequency of the harmonic numbers in the qpss analysis.

For example, assume that the qpss has a 1GHz LO input, and a 900MHz RF input. You want a log sweep above the first harmonic of the 1GHz signal in the qpss analysis. In this case, you need to select *relative* sweep, and specify 1 0 in the *Relative Harmonic* field. The first number is the frequency shift in multiples of the LO frequency, and the second number is the frequency shift in multiples of the RF input frequency. Next, type in 1K to 100M for the frequency range with a log sweep and 3 to 5 points per decade.

The noise analysis output frequency will start at 1G + 1K, and sweep to 1G + 100M using log spacing from 1K to 100M. This is illustrated below.

© 1994-2020
All Rights Reserved.

To use a log sweep below the first harmonic of the 1GHz input, use the same frequency range, and type in -1 0 as the relative harmonic. This is illustrated below.

Most of the time, the LO signal is the large signal in qpss. This is the first number in the shift. Generally, we do not design to deliberately mix with the RF blocking signal, so generally, the second number is zero. The first number is usually 1 or -1, because generally the RF signal mixes with the first harmonic of the LO. If the first number is -1, this is the signal just below the first harmonic of the mixer. If the first number is positive, it means the design input frequency is above the LO frequency.

Maximum Clock Order

In qpnoise terminology, the large signal from qpss is called the clock. Thus, *Maximum clock order* specifies the highest large signal harmonic number (The large signal is usually the LO signal) to calculate the conversion gain terms for. Usually, the conversion gain is needed for all the harmonics that are present in the qpss analysis, so usually *Maximum clock order* should be set to the number of LO harmonics in the QPSS analysis. Each conversion gain term that is calculated is called a sideband in a qpnoise analysis.

Sideband Numbers

The sideband number is the qpss harmonic number that is mixed with. Because qpss always has multiple input frequencies, qpss has multiple harmonic numbers that are being mixed with. If there are 2 tones in the qpss simulation, there are two numbers in qpnoise that

© 1994-2020 All Rights Reserved.

describe which harmonics of which tone are mixing to produce that output frequency. If there are three tones in the qpss analysis, then there are three numbers for each qpnoise sideband.

If the sideband number is negative, the noise frequency is more negative than the output frequency specified in the qpnoise *Choosing Analyses* form. If the sideband is positive, the noise frequency is more positive than the output frequency.

Although this seems complicated, there are features in the *Direct Plot Form* to make this easier to understand.

Conceptually, an infinite number of conversion gain terms exist when a single output frequency is analyzed in a nonlinear system. From the practical point of view, usually only the noise frequencies near all the qpss harmonics need to be measured.

Setting Moderate Harmonics in QPSS to get Accurate QPnoise Results

The qpss analysis, when shooting is selected, has a minimum of 200 timepoints in each slice, so the large signal in qpss inherently has frequency domain content through the 100th harmonic of the large signal input frequency. Only the harmonics that are specified in the moderate tone affect the accuracy of the solution. The result of this is that the qpnoise mixing products that are produced from mixing with the large signal are almost always correct, (Usually the LO tone) but the mixing with the moderate signal might or might not be accurate, depending on the number of harmonics that are specified for the moderate signal, and the number of harmonics actually produced by the system. Like harmonic balance, you need to run the qpss and qpnoise analyses with a number of harmonics specified for the moderate tone(s), and make the measurement in qpnoise. Then increase the number of moderate harmonics, run again, and make the same measurement. If the result changed significantly, then you need more moderate harmonics. If the result did not change, you might be able to reduce the number of moderate harmonics. Use the smallest number of moderate harmonics that produce a stable qpnoise results.

When harmonic balance is selected for the qpss engine, the adjustment of harmonic numbers above also needs to include the LO harmonics. Unlike shooting, where inherently the 100th harmonic is present in the qpss data for the large signal tone, only the number of harmonics actually specified for the LO are present in the solution.

Noise Separation

In addition to the total output noise, the individual noise contributions can be plotted if noise separation is selected in the *Choosing Analyses* form. More information about the noise separation will be provided in the examples section. The things that can be plotted are:

© 1994-2020 All Rights Reserved.

1. Total noise at the output from each individual noise input frequency (sideband). This allows the identification of which noise frequencies are causing the noise problem.
2. The noise at the output from each instance name with all noise mechanisms included for mixing from an individual noise input frequency. Once the noise frequency is identified in point 1 above, this capability plots the total noise from each component at the troublesome noise frequency. This allows the identification of which component in the circuit is causing the problem.
3. The noise at the output from each instance name with all noise mechanisms broken out separately for mixing from an individual noise input frequency. This capability identifies the specific noise sources at the troublesome noise frequency. For example, if a parasitic resistor was found to be the problem, the component might be resized to reduce the noise.
4. The current noise at the noise instance for each instance name with all noise mechanisms included for mixing from an individual noise input frequency. This capability has limited value. For this plot, all the noise source currents for all the noise mechanisms within the component are added together.
5. The noise current in the instance from each instance name with all noise mechanisms broken out separately for mixing from an individual noise input frequency. This capability allows the identification of which noise sources are the largest in the circuit. The measurement is at the source, not at the output.
6. The gain to the output from each individual instance with all the individual noise sources broken out separately for mixing from an individual input frequency. This capability plots the gain from the noise sources in point 5 above to the output of the circuit.

Input-Referred Noise

Important

When the qpnoise analysis is set up with the input source being a port, input-referred noise may not be exact. In this case, the input-referred noise is referred to the voltage source that is inside the port component. The amplitude of the noise is divided by two, which assumes a perfect match. If an accurate input-referred noise is desired, set the *Input Source* to *voltage*, and select the input source. This will disable the noise figure calculation.

© 1994-2020 All Rights Reserved.

ADE Implementation

The shift now is from theoretical to practical with the description of where you can find the settings talked about in the concepts section. This section describes the qpnoise *Choosing Analyses* form and the options form. For examples with all the steps shown, see the examples section that follows this section.

© 1994-2020 All Rights Reserved.

Below is the *Choosing Analyses* form.

© 1994-2020 All Rights Reserved.

1. The default *Sweeptype* is *absolute*, which means that the noise frequencies defined in the Fr*equency Sweep Range* fields will not be moved in frequency. The frequency sweep range is always an **output** frequency range.
2. Generally, it is better to select Linear or Log spacing and define your own sweep. Auto usually takes more points that are necessary, and that produces longer runtimes than needed.
3. *Maximum Clock Order* should be set to the same number that you have set the Large signal harmonics to in the qpss form. If you use harmonic balance as the engine in qpss, you can leave this field blank, which accomplishes the same thing.
4. When *Output* is set to *probe*, select a resistor or a port that serves as the load. The noise of this component will be subtracted for the noise figure calculation. That port or resistor also defines the output nets in the circuit. The nets that the port or resistor are connected to will be the output nets for the noise calculation.
5. When output is set to voltage, select a net or nets in the circuit as the output. No noise will be excluded from the noise figure calculation.
6. In order to make a noise figure calculation, the input must be a port.
7. When input-referred noise is needed, do not use a port as the input source. Instead, use a separate voltage source with a resistor in series. Although using a port does work, note that the input-referred noise is really referred to the voltage source that is in the port, and then that noise is divided by two in amplitude. This assumes a perfect match, which rarely occurs in practice.
8. The reference sideband defines the passband frequency for the noise figure calculations. The easiest way to define the passband frequency is to choose *Select from list*, and then select the input frequency range that defines the passband from the list. This also defines the input frequency for the input-referred noise calculation.
9. Noise separation is provided so that troublesome noise frequencies and components can be identified in the circuit. There is an example in the Example section below.

© 1994-2020 All Rights Reserved.

Qpnoise Options

None of the qpnoise options are commonly used.

tolerance

Leave this option at the default.

Qpnoise uses an iterative solver to calculate the noise, and any iterative solver needs a tolerance term to specify when the system has been solved accurately enough. The default for shooting qpnoise is 1e-9. The harmonic balance qpnoise tolerance is 1e-6.

gear_order

Do not set this option.

© 1994-2020 All Rights Reserved.

solver

The default is *turbo*.

When shooting is used as the engine in qpss, sometimes you will see warning messages in qpnoise that the residual is larger than the tolerance specified. This usually occurs at low offset frequencies from one of the harmonics in the qpss analysis. If you see this message, then select the *std* solver, which is more able to handle low offset frequencies. The runtime will increase when the *std* solver is selected.

When harmonic balance is the qpss engine, do not change this option.

resgmrescycle

Leave this option at the default, which is *short*.

hbprecond_solver

Note that if you plan to use harmonic balance qpss, the hb form is suggested instead.

This option is only available when harmonic balance is set for the qpss engine. When APS is not used, only the basic solver is available. When APS is used, first a faster preconditioner is used. Occasionally, this preconditioner will stagnate, and when it does, it issues a warning that it is switching to the basic solver, and it reverts back to the basic solver. A preconditioner is a mathematical algorithm that makes the iterative matrix solving process faster.

annotate

This option controls the level of detail in the output log. The default is *status*. The leftmost selection provides the least output. As you move right, more detail is provided in the output log.

save

The default is *allpub*. This saves all the public voltages at all levels of the hierarchy in the schematic. It excludes the internal nodes of the device models. *all* adds the internal nodes of all the devices. *lvl* saves all the nodes including the internal nodes of the devices through the level of hierarchy set in the *nestlvl* option. *lvlpub* is like *lvl*, but it does not save the internal nodes. *selected* saves only the nodes that are specifically saved.

© 1994-2020 All Rights Reserved.

nestlvl

If *save* is set to *lvl* or *lvlpub*, this controls the maximum level of hierarchy to be saved. If *nestlvl* is *1*, only the top level is saved. If *nestlvl* is 2, the top level and the next level down are saved. The value for *nestlvl* can be any integer.

saveallsidebands

Leave this option at the default value of *no*. If you want to see the noise contributors from the different noise input frequencies, select *yes* for noise separation in the qpnoise *Choosing Analyses* form.

additionalParams

This is typically used for customers who are evaluating new features. If you know the option name and the value, you can type it here. For example, you can type *solver=std* in this field instead of using the check box in the options form. multiple *keyword=value* pairs can be entered in this field. Separate the entries with a space.

© 1994-2020 All Rights Reserved.

Example

Noise Figure and Noise Summary

Consider a 2.4GHz LNA-Mixer.

© 1994-2020
All Rights Reserved.

The input source has variables set to a variable name. This is done in order to allow easy frequency and amplitude changes from the ADE environment without needing to change the schematic. Here is the properties list for the input port.

© 1994-2020
All Rights Reserved.

The variables are set in the ADE Explorer environment.

© 1994-2020
All Rights Reserved.

1. First, the qpss analysis needs to be set up. For more information, see the qpss section at the beginning of this chapter. Here, a 2.4GHz LO and a 2.45GHz blocker are set up.

© 1994-2020
All Rights Reserved.

2. Now set up the qpnoise analysis.

3. The output frequency range is set, and log sweep is used. It is usually better to set log or linear and define your own sweep because in general automatic takes more points than are usually needed.

© 1994-2020 All Rights Reserved.

4. *Maximum clock order* should be set to the number of harmonics that are used for the LO in the qpss *Choosing Analyses* form.
5. The output port and the input port are defined. Specifying *probe* for the *Output* and selecting either a port or a resistor automatically excludes the noise of the port from the noise figure calculation.
6. Using *Select from list* in the *Reference Side-Band* section is highly recommended. Select the passband frequency (the design input frequency) from the list.
7. If you want to see the different frequency and device contributions in the *Direct Plot Form*, select *yes* for *Noise Separation*.
8. Now run the analysis. When the analysis finishes, select *Results - Direct Plot - Main Form* from ADE Explorer. The *Direct Plot Form* is displayed.

9. In the *Direct Plot Form*, select qpnoise from the *Analysis* section.
10. Select the desired measurement, and click *Plot*.

© 1994-2020 All Rights Reserved.

11. If you have a noise figure spec over a defined bandwidth, select *Integrated Over Bandwidth*, type in the frequency range for the measurement, and click *Plot*. The noise figure curve, and the noise figure over the bandwidth are plotted.

Noise Summary

The noise summary provides information about the devices that contribute noise to the output. This information is available any time a qpnoise simulation completes.

In ADE Explorer, select *Results - Print - Noise Summary*. If noise separation is enabled, there will be two choices at the very top. `Qpnoise_src` has information about the noise currents at the individual noise sources in the circuit. `Qpnoise` has information about noise voltage at the output of the circuit.

If you want the noise contributors at a single frequency, select *spot noise*. If you want to have the noise integrated over a frequency range, select *integrated noise*.

If you want noise in volts, set the *noise unit* to *V.*

Generally, *Include All Types* should be selected. If you just want the noise from specific types of noise generators, you can select them from the list.

© 1994-2020 All Rights Reserved.

If you want to specifically include or exclude instances in the list, click the *Select* button to the right side of the *include instances* and *exclude instances* fields and then select the components in the schematic.

truncate just applies to the list. The total input and output-referred noise at the bottom of the noise summary output always includes all the noise from everything.

The figure below is the noise summary window with the sort function being set to noise contributors.

© 1994-2020
All Rights Reserved.

An example of the noise summary output sorted by noise contributors.

Results Display Window

Window Expressions Info Help cādence

Device	Param	Noise Contribution	% Of Total
/PORT1	rn	5.74846e-06	52.49
/I33/I0/NM10	id	1.93598e-06	5.95
/I33/I0/NM12	id	1.93598e-06	5.95
I33.I0.I28.I5.rs1	rn	1.23062e-06	2.41
I33.I0.I28.I11.rs1	rn	1.23062e-06	2.41
I33.I0.I28.I5.rs2	rn	1.2056e-06	2.31
I33.I0.I28.I11.rs2	rn	1.2056e-06	2.31
/I33/I0/R7/R3	rs	1.15988e-06	2.14
/I33/I0/R0/R3	rs	1.15988e-06	2.14
/I33/I1/I1/PM2	fn	1.05785e-06	1.78
/I33/I1/I1/PM4	fn	1.05784e-06	1.78
/I33/I1/I1/PM3	fn	1.05734e-06	1.78
/I33/I1/I1/PM5	fn	1.05719e-06	1.78
/I33/I1/I0/PM3	fn	9.5931e-07	1.46
/I33/I1/I0/PM5	fn	9.59303e-07	1.46
/I33/I1/I0/PM2	fn	9.5857e-07	1.46
/I33/I1/I0/PM4	fn	9.58564e-07	1.46
/PORT2	rn	8.17367e-07	1.06
I33.I0.I28.I5.rs11	rn	5.99171e-07	0.57
I33.I0.I28.I11.rs11	rn	5.99171e-07	0.57
I33.I0.I28.I5.rsbp1	rn	5.88183e-07	0.55
I33.I0.I28.I11.rsbp1	rn	5.88183e-07	0.55
I33.I0.I28.I5.rs22	rn	5.7922e-07	0.53
I33.I0.I28.I11.rs22	rn	5.7922e-07	0.53
I33.I0.I28.I5.rsbt1	rn	4.17919e-07	0.28

Integrated Noise Summary (in V) Sorted By Noise Contributors
Total Summarized Noise = 7.93427e-06
Total Input Referred Noise = 8.73444e-07
The above noise summary info is for qpnoise data

9

For all of the noise summaries, here is a table of the abbreviations used in the *Param* column:

Abbreviations Used In the Param Column

Resistor and Inductor:

Parameter	Meaning
`rn`	resistor thermal noise
`fn`	Flicker noise

© 1994-2020 All Rights Reserved.

Diode:

Parameter	Meaning
id	Diode current shot noise
rs	Series parasitic resistor noise

Gummel-Poon BJT:

Parameter	Meaning
fn	Flicker noise
ic	Collector current shot noise
ib	Base current shot noise
rc	Thermal noise from the collector parasitic resistor
rb	Thermal noise from the base parasitic resistor
re	Thermal noise from the emitter parasitic resistor

HiCUM:

Parameter	Meaning
ib	Base current shot noise
ic	Collector current shot noise
rbi	Thermal noise from the internal base parasitic resistor (See diagram below)
rbx	Thermal noise from the external base parasitic resistor (See diagram below)
rcx	Collector parasitic resistor thermal noise (See diagram below)
re	Emitter parasitic resistor thermal noise (See diagram below)
fn	Flicker noise

© 1994-2020 All Rights Reserved.

Figure 5-1 HiCUM Schematic Diagram

VBIC:

Parameter	Meaning
`rbi`	Thermal noise in resistor Rbi (See diagram below)
`rbx`	Thermal noise in resistor Rbx (See diagram below)
`rci`	Thermal noise in resistor Rci (See diagram below)
`re`	Thermal noise in resistor Re (See diagram below)
`rs`	Thermal noise in resistor Rs (See diagram below)
`itzf`	Collector current shot noise
`ibe`	Total base current shot noise

© 1994-2020 All Rights Reserved.

Parameter	Meaning
`ibex`	Node Bx to E base current shot noise (See diagram below)
`fn`	Ibe flicker noise (See diagram below)
`fnx`	Ibex flicker noise (See diagram below)
`iccp`	Parasitic transport collector to emitter current noise (See diagram below)
`ibep`	Parasitic transport base to emitter current noise (See diagram below)
`fnp`	Ibep flicker Noise (See diagram below)

© 1994-2020 All Rights Reserved.

Figure 5-2 VBIC Schematic Diagram

BSIM3:

Parameter	Meaning
`fn`	Flicker Noise
`id`	Thermal noise from the resistive drain current

© 1994-2020 All Rights Reserved.

Parameter	Meaning
rs	Thermal noise from the source parasitic resistor
rd	Thermal noise from the drain parasitic resistor

BSIM4:

Parameter	Meaning
fn	Flicker noise
id	Thermal noise from the resistive drain current noise
rs	Source parasitic resistor thermal noise
rd	Drain parasitic resistor thermal noise
igd	Gate-to-drain tunneling current noise
igs	Gate-to-source tunneling current noise
igb	Gate-to-bulk tunneling current noise
rbdb	Resistor thermal noise between dbNode and bNode (See diagram below)
rbpb	Resistor thermal noise between bNode and bNodePrime (See diagram below)
rbpd	Resistor thermal noise between Bulk and dbNode (See diagram below)
rbps	Resistor thermal noise between bulk and bNode prime
rbsb	Resistor thermal noise between sbNode and bNode (See diagram below)
rgbi	Gate Bias-Independent Resistor thermal noise

© 1994-2020 All Rights Reserved.

Figure 5-3 BSIM4 Schematic Diagrams

PSP-103

!

Parameter	Meaning
`sfl`	Channel current flicker noise
`sth`	Channel current thermal noise
`sig`	induced gate noise at 1 Hz

© 1994-2020 All Rights Reserved.

Parameter	Meaning
`sigth`	Induced gate noise
`shotgd`	Gate current shot noise from gate to drain
`shotgs`	Gate current shot noise from gate to source
`sjnoise`	Source to Bulk leakage current noise
`djnoise`	Drain to Bulk leakage and avalanche current noise
`rgatenoise`	Gate resistance thermal noise
`rbulknoise`	Bulk resistor thermal noise between node BP and BI (See diagram below)
`rjnoise`	Source side bulk resistor thermal noise between node BI and BS (See diagram below)
`rjundnoise`	Drain side bulk resistor thermal noise between node BI and BD (See diagram below)
`rwellnoise`	Well resistor thermal noise between node BI and B (See diagram below)
`rsourcenoise`	Source parasitic resistor thermal noise
`rdrainnoise`	Drain parasitic resistor thermal noise

Figure 5-4 PSP-103 Schematic Diagram

© 1994-2020
All Rights Reserved.

EKV 3.0

Parameter	Meaning
`fn_id`	Channel current flicker noise
`fn_ig`	Gate current flicker noise
`rs`	Source parasitic resistor thermal noise
`rd`	Drain parasitic resistor thermal noise
`rg`	Gate parasitic resistor thermal noise
`rsb`	Thermal noise for the substrate resistor connected between BSi and B. (See diagram below)
`rdb`	Thermal noise for the substrate resistor connected between BDi and B. (See diagram below)
`rbbs`	Thermal noise for the substrate resistor connected between BSi and Bi. (See diagram below where it is labeled $R_{DSB}/2$)
`rbbd`	Thermal noise for the substrate resistor connected between BDi and Bi. (See diagram below where it is labeled $R_{DSB}/2$)
`id`	Thermal noise of the channel resistor
`cor`	Induced gate noise
`shot_ig`	Gate current shot noise

© 1994-2020 All Rights Reserved.

Figure 5-5 EKV3.0 Schematic Diagrams

For other model types:

1. In a unix/linux shell window type `which spectre`.

© 1994-2020 All Rights Reserved.

This should return a path that ends in `tools/bin/spectre`.

2. Triple-click the line with the path, and then click the center mouse button to enter it as a command.
3. Use the backspace key, and delete the word `spectre`.
4. Type `cdnshelp` at the end of the expression, and press `Enter`.

   A browser window will be displayed.
5. Close the *Tip of the day* window.
6. At the top, you should see the release number of the simulator you are using.
7. Click the plus sign (+) to the left of the MMSim folder.
8. Click the plus sign (+) to the left of the *Spectre Circuit Simulator Components and Device Models Reference* folder.
9. Select the device type you are using.
10. At the bottom of that device listing, you will see *Component Statements*. Double-click on this.
11. Enter the name of the parameter in the search box in the right pane of the cdnshelp window.
12. Click the down arrow to the right of the *Find* field until you see the parameter name in the pane on the right side, and then see the definition of that parameter.

© 1994-2020 All Rights Reserved.

Viewing Noise Separation Results

Consider a 2.4GHz LNA-Mixer.

© 1994-2020
All Rights Reserved.

The input source has variables set to a variable name. This is done in order to allow easy frequency and amplitude changes from the ADE Explorer environment without needing to change the schematic. Here is the properties list for the input port.

© 1994-2020
All Rights Reserved.

The variables are set in the ADE Explorer environment.

© 1994-2020
All Rights Reserved.

1. First, the qpss analysis needs to be set up. For more information, see the qpss section at the beginning of this chapter. Here, a 2.4GHz LO and a 2.45GHz blocker are set up.

© 1994-2020
All Rights Reserved.

2. Now set up the qpnoise analysis.

© 1994-2020
All Rights Reserved.

3. The output frequency range is set, and log sweep is used. It is usually better to set log or linear and define your own sweep because in general automatic takes more points than are usually needed.
4. *Maximum clock order* should be set to the number of harmonics that are used for the LO in the qpss *Choosing Analyses* form. If you use the harmonic balance engine in qpss, this field can be left blank, which does the same thing.
5. The output port and the input port are defined. Specifying probe for the output and selecting either a port or a resistor automatically excludes the noise of the selected component from the noise figure calculation.
6. Using *Select from list* in the *Reference Side-Band* section is highly recommended. Select the passband frequency (the design input frequency) from the list.
7. If you want to see the different frequency and device contributions in the *Direct Plot Form*, select the *Noise Separation* check box.

© 1994-2020 All Rights Reserved.

8. Now run the analysis. When the analysis finishes, select *Results - Direct Plot - Main Form* from ADE Explorer. The *Direct Plot Form* is displayed.

Viewing the Total Noise at the Output from a Single Noise Input Frequency

This capability allows the noise from all sources in the circuit at different noise input frequencies to be measured. For example, all the noise that does not change frequency (the zero, zero sideband) can be plotted, along with the other noise input frequencies (sidebands) This plotting capability allows problem frequencies to be identified.

© 1994-2020 All Rights Reserved.

On the Direct Plot Form:

1. Select q*pnoise separation* from the *Analysis* section.
2. To view the total noise at the output select *Sideband Output* from the *Function* section.
3. Select *V/sqrt(Hz)* or *V**2/Hz* from the *Signal Level* radio buttons.
4. Select the appropriate modifier from the *Modifier* section.
5. Select the sideband number form the *Output Sideband* list. Start with about the -3 X sideband that borders the -4 X sideband.

© 1994-2020 All Rights Reserved.

6. Use the slider to get to the boundary between the 0 X sideband and the -1 X sideband. Hold the Press and hold the `<Shift>` key, and click the last 0 X sideband. A lot of sidebands will highlight.

7. In some cases, you want to start with the 0 X sideband and go through the 3 X sideband.

8. Usually, above the third harmonic, little noise is mixed to the output frequency.

© 1994-2020 All Rights Reserved.

9. Click *Plot*.

10. Select the trace with the highest amplitude at high frequency. Note that this is the -1 0 sideband in this case. -1 0 means the noise input frequency is mixing with the first harmonic of the LO and not with the RF input, and the noise input frequency is more negative than the output frequency. The noise frequency is 1KHz to 100MHz minus 1 * (LO Frequency of 2.4GHz) or -2.5GHz to nearly -2.4 GHz.

© 1994-2020 All Rights Reserved.

11. Select the trace with the largest 1/F noise.

12. This is the 0 0 sideband. There is no mixing at all with either the LO or the RF, so the noise frequency range is from 1KHz to 100MHz.

13. For the rest of this example, the -1 0 sideband is used. For isolating 1/F noise contributors, the 0 0 sideband would be used. The steps are exactly the same for both cases.

Viewing Total Noise at the Output from each Instance

Once the problem frequencies are identified in the plot of *Sideband Output*, the problem components need to be identified. From the previous plot, the -1 0 sideband contributes the largest amount of noise to the overall output noise. Using *Instance Output* allows the identification of which components with noise in the -1 0 sideband contribute the most noise. The total noise for all the noise mechanisms within the component is plotted.

© 1994-2020 All Rights Reserved.

On the *Direct Plot Form*:

© 1994-2020
All Rights Reserved.

1. Select *Instance Output* from the *Function* section.
2. Select *V/sqrt(Hz)* or *V**2/Hz*.
3. Select the appropriate modifier.
4. Select the sideband number. In this example, the -1 0 sideband was determined to contribute the most noise at the output of the circuit.
5. Select the type of component to display. If you want all of the types, choose *Include All Types*.
6. Specify the number of results you want to plot. 5 is specified in the above figure.
7. Click *Plot*.

   The total noise at the output for the top five components is plotted.

To see which component goes with which trace, select the plus sign (highlighted in the red box on the previous graphic) to the left or the text at the upper left of the display area. The legends expand, and you can see which trace is which in the *Name* column on the left side of the display tool.

© 1994-2020
All Rights Reserved.

The red trace above comes from the input port. The next largest contributor from the circuit is the green curve, which is selected. The instance name is shown in the legend. I33.I0.NM12 is the highest contributor. I33 is the top-level instance. I0 is the next level down instance. NM12 is the next level down instance.

Viewing the Exact Noise Mechanism Within the devices

The sideband with the largest contribution was determined by selecting *Sideband Output* and plotting. The specific component was identified by selecting *Instance Output*. The specific noise mechanisms can be identified using *Source output*. This plots the largest individual noise mechanisms for the sideband you select.

© 1994-2020 All Rights Reserved.

On the *Direct Plot Form:*

© 1994-2020
All Rights Reserved.

1. Select *qpnoise separation* from the *Analysis* section.
2. Select *Source Output* from the *Function* section.
3. Select *V/sqrt(Hz)* or *V**2/Hz*.
4. Select the appropriate modifier.
5. Select the sideband number.
6. Select the type of component to display. If you want all of the types, choose *Include All Types*.
7. If you want to restrict to one device, put its instance name in the *include inst* field. A list of devices separated by a space is also acceptable. In this case, we already know the instance name is I33.I0.NM12. This is entered into the Include Inst field above.
8. If you want to exclude a device (or devices) enter the instance name(s) in the *exclude inst* field.
9. Specify the number of results you want to plot in the *by top* field.
10. Click *Plot.*

© 1994-2020 All Rights Reserved.

The exact noise mechanism within the devices is plotted.

To see which noise source goes with which trace, select the plus sign to the left or the text at the upper left of the display area. The legends will be shown individually in the *Name* field of the waveform tool. The device name and the individual noise contributor will be shown. You can select either the individual legend or the trace and both the legend and the trace will be highlighted. The first part is the instance name, and the second part is the specific noise mechanism within that instance. For a list of the abbreviations used for the noise parameter, refer to Abbreviations Used In the Param Column.

Viewing the total noise currents within the devices

This capability has limited value. All the noise currents from all the noise mechanisms within the instance are added up without regard to the transfer functions to the output. No gain function is applied or are available for this measurement. It just adds up everything within that instance. No correlations are observed. It just adds up the total current.

© 1994-2020
All Rights Reserved.

In the *Direct Plot Form*:

© 1994-2020
All Rights Reserved.

1. Select *Instance source*.
2. Select *A/sqrt(Hz)* or *A**2/Hz* in the Signal Level selection.
3. Select *Magnitude* or *dB20* in the *Modifier* section.
4. Select the sideband number.
5. If you want every noise source to be considered, select I*nclude All Types*.
6. To exclude instances, click *Select* to the right of the *exclude inst* field, and select the device instances you want to exclude from the plot.
7. To include specific instances, click *Select* to the right of the *include inst* field, and select the instances on the schematic.
8. Set the number of curves to plot by setting the Truncate function.
9. Click *Plot*.
10. The frequency response curves are displayed.

11. The instance names are shown in the legend.

© 1994-2020
All Rights Reserved.

Plotting the individual noise source currents

This capability plots the individual noise currents within the device. This capability gives you the noise current at the source instead of at the noise at the output of the circuit. This capability allows you to identify the largest noise sources in the circuit.

© 1994-2020
All Rights Reserved.

1. Select *Primary Source*.
2. Select *A/sqrt(Hz)* or *A**2/Hz* in the *Signal Level* selection.
3. Select *Magnitude* or *dB20* in the Modifier section.
4. Select the sideband number.
5. If you want every noise source to be considered, select I*nclude All Types*.
6. To exclude instances, click *Select* to the right of the *exclude inst* field, and select the device instances you want to exclude from the plot.
7. To include specific instances, click *Select* to the right of the *include inst.* field, and select the instances on the schematic.
8. Set the number of curves to plot by setting the Truncate function.
9. Click *Plot*.
10. The frequency response curves are displayed
11. The specific noise sources are displayed in the legend.

For a list of the noise parameter names, Abbreviations Used In the Param Column. In this list, *rn* is a parasitic resistance, and *id* is the thermal noise from the channel.

© 1994-2020 All Rights Reserved.

Plotting the gain from the individual noise sources to the output

This capability gives the gain from the individual noise currents to the output of the circuit.

© 1994-2020
All Rights Reserved.

1. Select *Source Noise Gain*.
2. Select *A/sqrt(Hz)* or *A**2/Hz* in the *Signal Level* selection.
3. Select *Magnitude* or *dB20* in the *Modifier* section.
4. Select the sideband number.
5. If you want every noise source to be considered, select *Include All Types*.
6. To exclude instances, click *Select* to the right of the *exclude inst* field, and select the device instances you want to exclude from the plot. Everything will be plotted except for those instances.
7. To include specific instances, click *Select* to the right of the *include inst.* field, and select the instances on the schematic. All other instances will not be in the results when you use this capability. this example, the instance name of the largest noise source from the previous section was entered in the include inst field.
8. Set the number of curves to plot by setting the *Truncate* function.
9. Click *Plot*.

The frequency response curves are displayed. For a list of the abbreviations used for the noise parameters, Abbreviations Used In the Param Column. In this list, *fn* is flicker noise, and *rn* is resistor thermal noise.

© 1994-2020 All Rights Reserved.

Noise as a Function of Blocker Power

This measurement is almost always faster when harmonic balance is used. Consider using hb and hbnoise analysis as described in Chapter 3, “Frequency Domain Analyses: Harmonic Balance,” because the hb and hbnoise forms always include the latest improvements, and qpss and qpnoise generally are not improved. Consider a 2.4GHz LNA-Mixer.

© 1994-2020
All Rights Reserved.

The input source has variables set to a variable name. This is done in order to allow easy frequency and amplitude changes from the ADE Explorer environment without needing to change the schematic. Here is the properties list for the input port.

© 1994-2020
All Rights Reserved.

The variables are set in the ADE Explorer environment.

1. First, the qpss analysis needs to be set up. For more information, see the qpss section at the beginning of this chapter. Here, a 2.4GHz LO and a 2.45GHz blocker are set up. Set up a sweep of the blocker power that is appropriate for your circuit. In this example,

© 1994-2020 All Rights Reserved.

the blocker power is swept from -50 to -30dBm. Because the power will be swept to a large amplitude, 5 harmonics are set on the moderate tone.

© 1994-2020
All Rights Reserved.

2. Now set up the qpnoise analysis.

© 1994-2020
All Rights Reserved.

3. When the power is swept in qpss, the *Output Frequency Sweep Range* is set to *Single Point* automatically. Set the frequency to an appropriate output frequency for your design. In this example, 1MHz is set.

4. *Maximum clock Order* should be set to the number of harmonics that are used for the LO in the qpss *Choosing Analyses* form. If harmonic balance is used as the qpss engine, this field can be left blank, which has the same effect.

5. The output port and the input port are defined. Specifying Probe for the output and selecting either a port or a resistor automatically excludes the noise of the selected component from the noise figure calculation.

6. Using *Select from list* from the *Reference Side-Band* drop-down list is highly recommended. Select the passband frequency (the design input frequency) from the list. In this example, the input frequency just above the 2.4GHz LO is selected.

7. Click *OK* and run the analysis. When the analysis finishes, select *Results - Direct Plot - Main Form* from ADE Explorer. The *Direct Plot Form* is displayed.

.

8. Select *qpnoise* in the *Analysis* section.

© 1994-2020 All Rights Reserved.

9. Select the desired measurement and click *Plot*.

In this example, the double-sideband noise figure is plotted

The output appears in the waveform tool.

© 1994-2020
All Rights Reserved.

10. Next, the display will be made easier to read. Move your mouse cursor over one of the numbers on the X Axis, click the right mouse button and select *Swap Sweep Var* from the context menu.

11. In the *Swap Sweep Var* window, the sweep variable is selected automatically. Type in the X-Axis value for the measurement. In this example, this is 1M.

12. Click *OK*.

© 1994-2020
All Rights Reserved.

The waveform window adds a subwindow with a more traditional presentation.

13. As expected, the noise figure increases with increasing blocker power.

© 1994-2020
All Rights Reserved.

QPXF Analysis

Usually, qpxf is used to measure forward conversion gain when an LO and a high amplitude blocking signal are applied to the circuit. In this application, hb with hbxf is likely to run faster than shooting qpss with qpxf. Although harmonic balance is available in qpss, it is strongly recommended that you use hb and hbxf because the latest capabilities in the simulator are always available there.

For more information, see Periodic Transfer Function Analysis (PXF) on page 1117.

Qpxf is just like pxf in that qpxf calculates forward conversion gain. The difference is that qpxf runs after a qpss analysis where there are more input frequencies than in pss, and as a result, the circuit produces many more harmonics. Because there are many more harmonics in the large-signal analysis, there are many more transfer functions to calculate. Unlike pxf, qpxf has only one mode which measures the conversion gain of the system. If you need to measure AM to PM conversion (using modulated) or an instantaneous transfer function (using sampled) you need to use pss-pxf which has these capabilities. Qpxf does not have these capabilities.

As engineers, we are accustomed to driving the input and measuring the output of the circuit. This is the idea of qpac. In qpac, we apply the LO signal (or the sample clock) and a blocking signal in the qpss analysis, and then we drive the input with the small-signal qpac analysis and measure the mixing products that come out of the circuit. While this is very intuitive, the output display can be very confusing to read when the conversion gain near several qpss harmonics is desired. Qpxf changes the focus to the output. One output frequency at a time is considered, and in qpxf, the input frequencies that cause the output frequency are calculated, and the forward conversion gain from all those input frequencies is also calculated. Using qpxf, you can measure the conversion gain near a number of LO harmonics (or clock harmonics) and blocker harmonics in the same simulation run.

© 1994-2020 All Rights Reserved.

Example: Conversion Gain

Consider the double-balanced diode mixer shown below.

The RF input source has the frequency and amplitude set to variable names in order to allow changing the test setup without needing to change the circuit. This is suggested for all the sources used in the circuit that might need to be changed. This allows changes in frequency or amplitude from the ADE Explorer environment without needing to change the schematic itself.

© 1994-2020
All Rights Reserved.

The property list is shown below.

© 1994-2020
All Rights Reserved.

The LO signal also uses variable names.

The easiest way to get the variables list into ADE Explorer is to select *Variables - Copy From Cellview*. To set the values, select the value field to the right of the variable, type in the desired value, and press *Enter.* When you have set the variables, if you want to add the values back into the cellview, in the ADE Explorer window, select *Variables - Copy To Cellview*, and then perform a *Check and Save* in the schematic.

In this example, the LO frequency is 1GHz and its amplitude is 1.4 volts peak. This is +13 dBm, which is a common drive level for diode mixers. The RF frequency is 900MHz, and the amplitude is set to -10dBm.

The basic strategy is to apply the signals that cause the nonlinearity to occur in the qpss analysis, and then measure the forward conversion gain using qpxf analysis.

© 1994-2020 All Rights Reserved.

The qpss analysis needs to be set up first. The example below shows an LO signal at 1GHz with 10 harmonics, and the RF input at 900MHz with five harmonics. For more information, see the qpss section at the beginning of this chapter.

© 1994-2020 All Rights Reserved.

Next, the qpxf analysis needs to be set up. Like xf, pxf, and hbxf, the frequency range specified in the *Choosing Analyses* form is the **output** frequency range. Usually, it is better to select Linear or Log spacing and define the resolution of the sweep yourself. Most of the time, Automatic will take more frequency points than you need, and that extends the runtime.

When shooting is used as the qpss engine*, Maximum clock order* should be set to three or four. Usually, the higher order conversion gain products are not needed. If harmonic balance is used for the qpss engine, either leave this field blank, or set *Maximum clock order* to the same number you used for the LO harmonics. In either case, the *Direct Plot Form* can be used to select the result that you want.

Select the output net in the circuit. The conversion gain terms will be measured to this net. If you have an output current, add an iprobe from analogLib in series with the output, and select

© 1994-2020 All Rights Reserved.

probe in the *Output* section. Then specify the instance name of the iprobe in the *Output Probe Instance* field.

© 1994-2020
All Rights Reserved.

Once the analyses are set up, the simulation can be run. When the simulation finishes, select *Results - Direct Plot - Main Form* in the ADE Explorer window. The *Direct Plot Form* is displayed.

© 1994-2020
All Rights Reserved.

The example below shows a typical plot

.

To measure the conversion gain, place a marker at the desired **input** frequency, and read the forward conversion gain. The conversion gain is about -6dB as expected.

© 1994-2020
All Rights Reserved.

In a zoomed-in view, the conversion gain near the first harmonic of the LO (1GHz) is about 6 dB. The conversion gain near the third harmonic of the LO is also quite high at about -18.5 dB.

The qpxf direct plot is a bit non-intuitive. The *Choosing Analyses* form specifies the output frequency range and the output node. Qpxf calculates the transfer functions from all the sources in the circuit to that output node. Therefore, to plot, you select a source that you want the forward conversion transfer functions to be displayed. When the source in the schematic is selected, the waveform is displayed.

© 1994-2020 All Rights Reserved.

Note that all the short line segments are the forward conversion gain from the selected source (the LO source in the case below) to the output. None of the conversion gain terms are very high from the LO port to the output.

Overview of Simulation Capabilities

The primary application of qpxf is to measure conversion gain. Qpxf is a small-signal analysis. It calculates the small-signal conversion gain. If you need a large-signal measurement of conversion gain, then use a large-signal analysis like harmonic balance (hb) or quasi-periodic steady-state (qpss) analysis. Qpxf measures gain in a way that requires some thought for most engineers because we are so accustomed to connecting an input, driving the system, and measuring what comes out. This is the idea of qpac, where we capture the large-signal operating point in qpss, and then drive a small-signal input, and measure the mixing products that come out. If you think of this methodology, to measure the conversion gain around several harmonics usually requires a separate simulation near each harmonic.

Qpxf was created to allow the measurement of the conversion gain transfer functions near all the qpss harmonics at the same time. In qpac, we had one input frequency at a time, and output mixing products at multiple output frequencies are calculated. In qpxf, we focus on one

© 1994-2020 All Rights Reserved.

output mixing product at a time, and measure the input frequencies that will cause that output frequency, and then calculate the forward conversion gain from all those input frequencies to the single output frequency. Multiple output frequencies are usually analyzed in a single qpxf analysis. Unlike qpac where the output can be seen at every node in the circuit, qpxf calculates the gain only from all the independent sources in the circuit to the output. Thus, to measure a conversion gain, you select the source instance in the schematic from the Direct plot form instead of selecting a net.

For example, imagine a mixer with an LO frequency of 1GHz and an RF signal at 900MHz. (Think easy math.) Now, imagine the down conversion case where the output is 1MHz. Given the LO frequency, inputs at 1MHz, 999MHz, 1.001GHz, 1.999GHz, 2.001GHz, 2.999GHz, 3.01GHz, and so on all cause an output at 1MHz. Signals 1MHz on either side of the RF harmonics will also convert to 1MHz, so 899MHz, 901MHz, 1799MHz, 1801MHz and so on will have conversion gain terms. There are also intermodulation products that are created by the circuit, and because there are so many conversion gain terms, it can be a bit confusing. Depending on the circuit, all those different input to output paths will have different gains from each input frequency to the output at 1MHz. In qpxf, you specify the highest qpss large signal harmonic to calculate the transfer functions from in the Maximum clock order field in the Choosing Analyses form. All these different input frequencies are called sidebands in qpxf. There is the ability to plot just the desired conversion gain terms in the Direct Plot form.

Frequency Sweep

This analysis is like xf and pxf, where the **output** frequency is varied. The output frequency range to be analyzed is specified in the qpxf *Choosing Analyses* form. Qpxf then calculates the conversion gain terms.

In addition to the frequency, there is a selection called *Sweep Type*. The choices are *relative* and *absolute*. When *absolute* is selected, the frequency range is used directly with no frequency conversion. When *relative* is selected, a *Relative Harmonic* field appears. The frequency sweep is shifted by the specified frequency of the harmonics in the qpss analysis. For example, assume that the qpss has signals at 1GHz and at 1.1GHz, and a log sweep is desired above the first harmonic of the 1GHz signal in the qpss analysis. In this case, you need to select *relative* sweep, and specify 1 0 in the *Relative Harmonic* field. Next, type in 1K to 100M for the frequency range with a log sweep and 3 to 5 points per decade. The first number is the frequency shift in integer multiples of the Large signal in qpss, which is usually the LO frequency. The second number is the frequency shift in integer multiples of the RF blocker frequency. Since usually we design to mix with the first harmonic of the LO and not with the RF signal, the first number is usually either 1 or -1, and the second number is usually zero.

© 1994-2020 All Rights Reserved.

Assuming the qpss analysis has inputs at 1GHz and 1.1GHz, and the qpxf frequency range is 1K to 100M, and the relative harmonic number is 1 0, the sweep will start at 1G + 1K, and sweep to 1G + 100M using log spacing from 1K to 100M. This is illustrated below.

To use a log sweep below the first harmonic, use the same frequency range, and type in -1 0 as the relative harmonic. This is illustrated below.

© 1994-2020
All Rights Reserved.

Maximum Clock Order

In qpxf terminology, the large signal from qpss is called the clock. Thus, *Maximum clock order* specifies the highest large signal harmonic number (The large signal is usually the LO signal) to calculate the conversion gain terms for. Usually, the conversion gain is only needed for a small number of LO harmonics, so when shooting is used as the qpss engine, usually *Maximum clock order* is fairly small, on the order of 3 to 5. Each conversion gain term that is calculated is called a sideband in a qpxf analysis. When harmonic balance is used for the qpss engine, the *Maximum clock order* field should be left blank, or the same as the number of LO harmonics, which does the same thing. The *Direct Plot Form* has the ability to select the individual terms that are needed.

The sideband number is the qpss harmonic number that is mixed with. Because qpss always has multiple input frequencies, qpss has multiple harmonic numbers that are being mixed with. If there are two input tones in the qpss simulation, there are two numbers in qpxf that describe which harmonics of which tone are mixing to produce that output frequency. If there are three tones in the qpss analysis, then there are three numbers for each qpxf sideband.

If the sideband number is negative, the input frequency is more negative than the output frequency specified in the qpxf *Choosing Analyses* form. If the sideband is positive, the input frequency is more positive than the output frequency.

Although this seems complicated, there are features in the Direct Plot form to make this easier to understand.

Setting Moderate Harmonics for Accurate QPXF Results

The qpss analysis, when shooting is selected, has a minimum of 200 timepoints in each slice, so the large signal in qpss inherently has frequency domain content through the 100th harmonic of the large signal input frequency. (Usually the LO signal) Only the harmonics that are specified in the moderate tone affect the accuracy of the solution. The result of this is that the qpnoise mixing products that are produced from mixing with the large signal are almost always correct, but the mixing with the moderate signal might or might not be accurate, depending on the number of harmonics that are specified for the moderate signal, and the number of harmonics actually produced by the system. Like harmonic balance, you need to run the qpss and qpxf analyses with a number of harmonics specified for the moderate tone(s), and make the measurement in qpxf. Then increase the number of moderate harmonics, run again, and make the same measurement. If the result changed significantly, then you need more moderate harmonics. If the result did not change, you might be able to reduce the number of moderate harmonics. Use the smallest number of moderate harmonics that produce a stable qpnoise results. If the moderate signal is below compression, start with two or three harmonics on the moderate tone. If the signal is at compression, start with five harmonics. If the signal is above compression, start with seven harmonics.

© 1994-2020 All Rights Reserved.

Qpxf is usually preferred to qpac when you need to measure the conversion gain near all the qpss harmonics.

Two methods are generally used to plot the conversion gain terms. One method is to plot the terms individually from the *Direct Plot Form*. Because there are so many harmonics to mix with and so many transfer functions, this method is preferred by many. A detailed example is in the examples section.

In the example below, the LO is at 1.9GHz, and the RF blocker is at 1.95GHz. The circuit is an image reject mixer with a 100MHz IF. Given this, the passband and image are at 1.8GHz and 2GHz. After the simulation has been run, in the *Direct Plot Form*, select *sideband* in the *Sweep* section. This allows individual conversion gain terms to be plotted. For the qpxf analysis, the output frequency range is 200KHz to 200MHz. Below is the *Direct Plot Form* for the low side injection transfer function. When the input frequency range is chosen, select the source in the circuit that you want the conversion gain terms for.

© 1994-2020 All Rights Reserved.

Below is the direct plot form for the high side injection transfer function. After the first selection, you can select *Replot*, which will take the same source as the choice for plotting.

© 1994-2020
All Rights Reserved.

Below are the transfer functions. Markers A and B have been placed at the passband and image frequencies.

The passband gain is 5.53dB, and the image rejection is about 63.4dB.

The other method (plotting all the transfer functions at the same time) is given in the examples section.

Qpxf can also be used to measure the conversion gain from the battery that supplies power to the output. All the gain and conversion gain terms should be below unity, or a potential

© 1994-2020 All Rights Reserved.

oscillator is created. In the plot below, the gain from the bias source to the output at the IF frequency of 100MHz is about 10dB. This circuit is a potential oscillator.

© 1994-2020
All Rights Reserved.

ADE Implementation

This section jumps from the theoretical to the practical with examples of how to use the qpxf analysis we just covered. The focus is only on the qpxf Choosing Analysis form so you can see where the individual settings are made, and the qpxf options form. For examples with all the steps shown, see the examples section that follows this section.

QPXF for a Normal Conversion Gain Measurement

Here is an example of the *Choosing Analyses* form for a normal conversion gain measurement.

© 1994-2020
All Rights Reserved.

Sweeptype can be relative or absolute. The default is absolute. The selection that is being used is shown to the right of the *Sweeptype* selection.

1. The frequency sweep range is always an **output** frequency range. It is not a sweep in the traditional sense of connecting an input, sweeping over a frequency range, and measuring the output. Instead, it is an output frequency range to be analyzed. At each output frequency in the sweep, the input frequencies that will mix with the qpss harmonics to cause that output frequency are calculated, and the forward conversion gain is also calculated.

2. Linear and log spacings are provided. Generally, Auto should be avoided because it will always calculate 50 frequency points. Usually, a much smaller number is needed, so it is recommended that you set Linear or Log, and control the resolution yourself to save time.

3. Although there are choices for the Sidebands selection, it is usually better to leave the setting at *Maximum clock order*. When you choose shooting as the qpss engine, and if you need a small subset of the output results, you can set a small number (like 3 or 5) in the *Maximum clock order* field. If you use harmonic balance, set *Maximum clock order* to the same number you used to set the LO harmonics in qpss, or leave the field blank, which accomplishes the same thing. In either case, you can use the *Direct Plot Form* to select just the results you want.

4. Run the simulation, and make the measurement in qpxf. Now increase the number of moderate harmonics by about 50% and run again. If the qpxf result changed significantly, then you need more harmonics on the moderate tone (or tones) in qpss. If it did not change, use the smallest number of harmonics that results in an stable measurement.

5. The output can measure either a voltage or a current. The voltage measurement is the default. Specify the output node or nodes in the circuit. If the output is a current select probe for the output, add an iprobe from analogLib (a current probe) in series with the output in the schematic, and specify the instance name in the *Output Probe Instance* field in the *Choosing Analyses* form.

© 1994-2020 All Rights Reserved.

QPXF Options

Below is the options form.

None of the qpxf options are commonly used.

tolerance

Leave this option at the default value.

Qpxf uses an iterative solver to calculate the conversion gain terms. Any iterative solver needs an error tolerance to specify when to stop iterating because the solution is accurate enough. and The tolerance option specifies that accuracy for qpxf. For shooting, the default tolerance is 1e-9. When HB is the qpss engine, the default is 1e-6.

© 1994-2020 All Rights Reserved.

gear_order

Do not change this option.

solver

The default solver is the turbo solver.

When hb is used as the engine in qpss, leave this option at the default.

When shooting is used, sometimes when the qpxf input frequency is very close to the frequency of one of the harmonics in the qpss, warning messages will appear in the qpxf output warning that the accuracy might not be good enough. If you see these messages, select the *std* solver, which has better ability to handle frequencies that are very close to a harmonic in the qpss, but which takes longer to run than the *turbo* solver.

resgmrescycle

Leave this option at the default.

hbprecond_solver

This option is only available only when harmonic balance is selected in qpss and APS is used.

The basic solver is the only solver available in standard Spectre. Autoset is the default solver when APS is used. This solver is faster, but occasionally stagnates. When stagnation is detected, APS automatically switches to the basic solver, and prints a message in the Spectre output window. If you have stagnation, it will save a small amount of time to set this option to *basicsolver*.

annotate

This option controls the level of detail in the Spectre output log. No detail is provided when you select *no*, while more details are provided as you move towards right with steps option providing maximum detail.

stimuli

This option is not implemented at the current time.

© 1994-2020 All Rights Reserved.

freqaxis

freqaxis specifies whether you want to see the negative frequency axis or not.

The qpxf analysis calculates the input frequencies and forward conversion gain, so *in* and *absin* are reasonable choices. *in* displays the negative frequency axis. *absin* (absolute value of the input) displays positive input frequencies. This is the default.

save

This option is not implemented at the current time.

nestlvl

This option is not implemented at the current time.

additionalParams

additionalParams is typically used for new features that are being beta tested. Keyword=value pairs are expected in this field. An example is `solver=std`.

For more information about the other options, type `spectre -h qpxf` at the command prompt in a Unix shell window.

© 1994-2020 All Rights Reserved.

Example

Note that for this application, harmonic balance is likely faster than shooting. If you plan on using harmonic balance, please use the hb and hbnoise analyses to measure the transfer functions. Hbnoise has hbxf integrated into the noise measurement. The hb and hbnoise Choosing Analyses forms always have the latest harmonic balance features.

Consider the LNA-Mixer circuit shown below.

© 1994-2020
All Rights Reserved.

1. The input port has variable names to set the values of the frequency and amplitude. This is suggested so these terms can be easily changed in the ADE Explorer environment without needing to change the schematic.

© 1994-2020
All Rights Reserved.

2. The variables are set in ADE Explorer.
3. The LO is at 2.4GHz and the RF blocker is at 2.5GHz.

© 1994-2020
All Rights Reserved.

4. First, a qpss analysis needs to be set up. For more information, see the qpss section at the beginning of this chapter.

© 1994-2020
All Rights Reserved.

5. Now set up the qpxf form.

6. Choose an appropriate output frequency range to analyze for your circuit. This frequency range is wider than usual so the conversion gain curves can be easily seen in the waveform tool.

7. Usually, it is better to select Linear or Log spacing, and define your own sweep. Automatic usually takes more points than needed, and this extends the runtime.

8. Set *Maximum clock order* as appropriate for your needs. In this case, mixing product that mix with the third harmonic and lower of the LO are calculated. If you use the harmonic balance engine in qpss, the *Maximum clock order* field should either be left

© 1994-2020 All Rights Reserved.

blank, or should be set to the same number of LO harmonics in the qpss analysis. Both methods accomplish the same thing.

9. Choose an output node in the circuit. If your circuit has an output current, add an iprobe (current probe) from analogLib in series with the output current, and specify it in the *Output* section.

10. Now run the simulation. When the simulation completes, select *Results - Direct Plot - Main Form* in the ADE Explorer window. The *Direct Plot Form* appears.

11. To plot individual conversion gain terms, select *sideband* in the *Sweep* section.

12. Select the **input** frequency range you want to plot in the *Input Sideband* section.

13. If you want this measurement to plot automatically after subsequent simulations, select *Add To Outputs* at the bottom of the *Direct Plot Form*. In this example, it was not selected.

© 1994-2020 All Rights Reserved.

14. Click the desired source in the schematic for the gain measurement. For the plot below, the input port was selected. To measure the conversion gain, move your mouse cursor to the input frequency, and type m to place a marker. The conversion gain at 2.45GHz input frequency is about 18dB.

15. There are also mixing terms from mixing with the RF blocker signal as well. When the output frequency is near zero to 400MHz, the input frequency is from 2.1GHz to nearly

© 1994-2020
All Rights Reserved.

2.5GHz. Select that term from the *Direct Pot Form*, and if you still want the gain from the input source, select *Replot*.

16. If you want this measurement to plot automatically after subsequent simulations, select *Add To Outputs* at the bottom of the *Direct Plot Form*. In this example, it was not selected.

© 1994-2020 All Rights Reserved.

17. The conversion gain from mixing with the RF input is quite small.

.

© 1994-2020
All Rights Reserved.

18. All of the conversion gain terms can also be plotted. To do this, select *spectrum* in the *Sweep* section.

19. If you want this measurement to plot automatically after subsequent simulations, select *Add To Outputs* at the bottom of the *Direct Plot Form*. In this example, it was not selected.

© 1994-2020 All Rights Reserved.

20. All of the conversion gain terms are plotted.

21. There are a lot of conversion gain terms because there are a lot of harmonics to mix with.

© 1994-2020
All Rights Reserved.

22. Now the waveform window is zoomed in to the part of the output where the gain is larger than -100dB. Select one of the curves. It highlights.

23. The sideband number is 1 0. The output frequency from the *Choosing Analyses* form is 1KHz to 400MHz. The first number is the frequency shift from output to input in multiples of the LO frequency. The second number is the frequency shift in multiples of the RF blocker frequency. The input signal can mix with any of the harmonics the system produces. 1 0 means that the input is mixing with the first harmonic of the LO and not with the RF input. Positive numbers mean that the input frequency is more positive than the output frequency. so, in this case, the input frequency is 1KHz to 400MHz + (1 * LO frequency of 2.4GHz) = 2.4+ GHz to 2.8GHz.

24. Below, the 1 -2 sideband is selected. The input mixes with the first harmonic of the LO and the second harmonic of the RF. 1KHz to 400MHz + (1 * 2.4GHz) + (-2 * 2.5GHz) = -

© 1994-2020
All Rights Reserved.

2.59999GHz to -2.2GHz. By default, this is reflected up into the positive frequency domain.

25. It is always suggested that you check the conversion gain from the power supply to the output to see if there are any mixing paths that have greater than unity gain. To measure

© 1994-2020
All Rights Reserved.

these conversion gain terms, select the DC source in the schematic that supplies power to the circuit. In this case, all the conversion gain terms are less than unity.

26. Note that finding a gain path like this in the lab is extremely difficult. This is one of the benefits of simulation.

© 1994-2020
All Rights Reserved.

Conversion Gain as a Function of Blocker Power

Consider the LNA-Mixer circuit shown below.

© 1994-2020
All Rights Reserved.

1. The input port has variable names to set the values of the frequency and amplitude. This is suggested so these terms can be easily changed in the ADE Explorer environment without needing to change the schematic.

© 1994-2020 All Rights Reserved.

© 1994-2020
All Rights Reserved.

2. The variables are set in ADE Explorer.
3. The LO is at 2.4GHz and the RF blocker is at 2.45GHz.

© 1994-2020 All Rights Reserved.

4. First, a qpss analysis needs to be set up. Specify a sweep of blocker power that is suitable for your system. For more information, see the qpss section at the beginning of this chapter.

© 1994-2020
All Rights Reserved.

5. Now set up the qpxf form, as show below.

6. The *Output Frequency Sweep Range* is set to *Single-Point* when a sweep is performed in qpss. Set the desired output frequency in the *Freq* field.

7. Set *Maximum clock order* as appropriate for your needs. In this case, mixing product that mix with the third harmonic and lower of the LO are calculated. If you are using the harmonic balance engine in qpss, this should be set to the number of LO harmonics in the qpss analysis, or left blank, which does the same thing.

© 1994-2020 All Rights Reserved.

8. Choose an output node in the circuit. If your circuit has an output current, add an iprobe (current probe) from analogLib in series with the output current, and specify it in the *Output* section.

9. Now run the simulation. When the simulation completes, select *Results - Direct Plot - Main Form* in the ADE Explorer window. The *Direct Plot Form* is displayed.

10. Select *variable* from the *Sweep* section.

11. Select the input frequency range you want to plot in the *Input Sideband* section. Note that the *Input sideband* field is labeled incorrectly as *Output Harmonic*.

12. If you want this measurement to plot automatically after subsequent simulations, click *Add To Outputs* at the bottom of the *Direct Plot Form*. In this example, it was not selected.

© 1994-2020 All Rights Reserved.

13. Click the desired source in the schematic for the gain measurement. For the plot below, the input port was selected.

14. As expected, the conversion gain drops with blocker power.

© 1994-2020
All Rights Reserved.

QPSP Analysis

In order to run qpsp, a qpss analysis needs to be run first. For many applications, qpss will run much faster using the harmonic balance engine. If you want to use harmonic balance, instead of using the qpss and qpsp *Choosing Analyses* forms, use the hb and hbsp forms instead. The hb form always has the latest harmonic balance features in it, and the qpss form usually does not.

Please see the section in Chapter 4 on psp analysis. Qpsp is just like psp. The only difference is that instead of running after a pss analysis, it runs after a qpss analysis. Conceptually, the only difference is that in qpss, there are many more harmonics, and because of that there may be mixing paths from the input frequency to the output frequency by mixing with different harmonics. This will be shown in the example below.

Qpsp is provided to measure S-Parameters for circuits that translate frequencies. All the basic assumptions of S-Parameters apply to qpsp, with the addition of having different input and output frequencies. Qpsp can be used for up or down-conversion applications.

© 1994-2020 All Rights Reserved.

Example: Down-Conversion Diode Mixer

Consider the double-balanced diode mixer circuit below:

The input source has the frequency and amplitude set to variable names in order to allow changing the test setup without needing to change the circuit. This is suggested for all the sources used in the circuit that might need to be changed. This allows changes in frequency or amplitude from the ADE Explorer environment without needing to change the schematic itself.

© 1994-2020 All Rights Reserved.

The property list is shown below.

© 1994-2020
All Rights Reserved.

The same strategy is used for the LO port.

The easiest way to get the variables list into ADE Explorer is to select *Variables -> Copy From Cellview*. To set the values, select the value field to the right of the variable, type in the

© 1994-2020
All Rights Reserved.

desired value, and press *Enter.* When you have set the variables, if you want to add the values back into the cellview, in the ADE Explorer window, select *Variables - Copy To Cellview*, and then perform a *Check and Save* in the schematic

.

The first RF input is at 900MHz, the second RF input frequency is disabled (by setting the frequency to zero), and the LO is at 1GHz with a 1.4 volt peak signal. This is +13 dBm, which is a common drive level for a diode mixer.

The basic strategy is to apply the signals that cause the nonlinearity in the qpss analysis, and then measure the frequency-shifted S-Parameters using qpsp. Doing this allows measuring the input match at the input frequency, and the output match at the output frequency in the same analysis. For the forward and reverse gain terms, the specified frequency translation is taken into account.

© 1994-2020 All Rights Reserved.

The qpss analysis needs to be set up first. The example below shows the LO input at 1GHz with 10 harmonics and the RF blocker input with 5 harmonics. For more information, see the qpss section at the beginning of this chapter.

Next, the qpsp analysis needs to be set up. The frequency range depends on your application, and can be either an input frequency range, or an output frequency range. This

© 1994-2020 All Rights Reserved.

is the input frequency range just above the LO frequency. The actual frequency for each port-frequency definition will be set up later.

© 1994-2020
All Rights Reserved.

Start with Port 1. In the *Port #* column field, type 1. In the *Name* column field, either type the instance name, or use the *Select Port* button just below the edit line to select the port in the circuit. In this case, Port 1 is the RF port.

© 1994-2020
All Rights Reserved.

Now click *Choose Harmonic*. In the *Choose Harmonic* window, select the frequency range you want associated with that port number. For the down-conversion application, The input is just above the LO frequency of 1GHz.

Note that there are multiple instances at the same input frequency range. This is because there are mixing paths using different harmonics that produce the same frequency range. Choose the entry where the sum of the absolute value of the indices is smallest. In this case, 0 + 0 = 0, and 4 + 2 = 6. Zero is less than six, so that is the entry that is chosen.

Like the other qpxx analyses, the sideband number displays the frequency shift for each input in multiples of the input frequency. In other words, a signal can mix with different harmonics of the input to provide an output.

In this case, 0 0, means do not mix with anything. The center frequency is 1.1GHz. For the other term, 1.1GHz -4*(1GHz) +2*900MHz = -1.1GHz. The default is to reflect this term to the positive frequency domain. The -4 2 term mixes down with the fourth harmonic of the LO and up with the second harmonic of the RF blocker.

Click *OK*.

© 1994-2020 All Rights Reserved.

Click *Add* to the right of the *Choose Harmonic* button. This adds that port and frequency definition to the list for the analysis. By convention, port 1 defines the input frequency.

Note that port 1 is now the input port at the input frequency range of 1.05GHz to 1.15GHz. A single port definition in qpsp has both a port name and a frequency definition.

In qpsp, multiple port and frequency definitions can be run at one time.

© 1994-2020 All Rights Reserved.

The next step is to measure RF to IF conversion gain. Type *2* in the *Port #* column field. Click the *Select Port* button, and select the IF port.

© 1994-2020
All Rights Reserved.

Now define the frequency range for that port. Click *Choose Harmonic*, and select the IF frequency from the list. Next, click *OK*.

Again, there are multiple entries at the same frequency from multiple mixing paths. Taking the center frequency of 1.1GHz, then one term is 1.1G - 1 * 1G = 100M. The other term is 1.1G -3 * 1G + 2 * 900M = -100MHz

Take the entry where the sum of the absolute values of the indices is smaller.

© 1994-2020 All Rights Reserved.

Click the *Add* button to the right of the *Choose Harmonic* button. The second port-frequency definition is added to the list. By convention, port 2 defines the output frequency range.

Now measure the RF to IF isolation. Type 3 in the *Port #* column field. Click *Select Port*, and again select the IF output port. Click *Choose Harmonic*, and select the input frequency

© 1994-2020
All Rights Reserved.

range. Click *OK*. Once again, there are multiple entries at the same frequency from multiple mixing paths. Take the entry where the sum of the absolute values of the indices is smaller.

© 1994-2020 All Rights Reserved.

Click *Add* to the right of the *Choose Harmonic* button. The third port-frequency definition is added.

Make sure all the port definitions are in the list. It is easy to forget to add the last port.

© 1994-2020
All Rights Reserved.

With the above port-frequency definitions, S_{11} measures the input match at the RF frequency. S_{22} measures the output match at the IF frequency. S_{31} measures the RF to IF isolation, and S_{21} measures the RF to IF conversion gain.

Once the analyses are set up, the simulation can be run. When the simulation finishes, select *Results - Direct Plot - Main Form* in the ADE Explorer window.

The example below shows a typical *Direct Plot Form*.

© 1994-2020
All Rights Reserved.

If you click *Add To Outputs*, an expression is entered under the Maestro tab of the ADE Explorer window so that the next time you simulate your circuit, this measurement plots automatically.

Here are the S_{11}, S_{21}, S_{31}, and S_{33} plots.

.

The conversion power loss is about 6.5dB. The input match is about -8dB. The output match is about -9.9dB. The RF to IF isolation is about -6.5dB.

© 1994-2020
All Rights Reserved.

Because *Add to outputs* was selected in the *Direct Plot Form* as shown previously, ADE Explorer now has expressions that will plot automatically on subsequent simulations.

Note that these can be plotted on a Smith Chart as well. The Smith Chart capability is only available from the *Direct Plot Form*. Expressions under the Maestro tab of ADE Explorer will always plot on a rectangular plot. The Smith chart has not been shown because this circuit is inherently very wide band. The curve on the Smith Chart would be tiny. An example using a Smith Chart is shown later on in the Examples section.

© 1994-2020 All Rights Reserved.

Qpsp can be used for up conversion as well. To do that, just change the frequency definitions for the ports. Since IF to RF gain is usually not needed, only a 2 port measurement has been set up. Note the frequency difference for Ports 1 and 2.

Note that the *Frequency Sweep Range* did not change. Only the frequency definitions for the two ports changed. With the above port-frequency definitions, S11 measures the input match at the low frequency.

© 1994-2020 All Rights Reserved.

S22 measures the output match at the upconverted frequency. S21 measures the small-signal conversion gain from low to high frequency.

Now run the analysis. S_{11}, S_{21}, and S_{22} are plotted below. The conversion power loss is about 6.5dB. The input match is about -8dB. The output match is about -9.9dB.

Note that the frequency range on the X Axis of the plot changed. This is because the input frequency changed, and by default, the input frequency is shown on the waveform plot X Axis.

© 1994-2020
All Rights Reserved.

Overview of Simulation Capabilities

Qpsp is similar to sp in that power gain measurements are made. Sp is based on the DC operating point, so it can only produce results that are linear. All the frequencies are the same for all the S-Parameter measurements.

Qpsp uses the qpss analysis results as the basis for making the S-Parameter measurement. Because the qpss result is the operating point solution for qpsp, the input can mix with any of the harmonics that are present in the qpss analysis for the frequency-translated terms. The term S_{XX} is not enough to describe the measurement. We need not only port numbers and names, but also the frequency needs to be defined for each port definition.

In the example in the preceding section, the LO was at 1GHz and the RF input was at 900MHz. The specified frequency for qpsp was just above the first harmonic of the LO. In the qpsp definition for each port, both the port name and the frequency that is associated with that measurement need to be defined.

Only the S-Parameters and the noise calculations are available for plotting in qpsp. Stability factors like Kf and the other calculations associated with sp like Gamma-Min are not available in qpsp.

Noise analysis is available in qpsp, and is the same as pnoise. Since very few people use qpsp for noise measurements, it is not covered in this section. Please see the pnoise or hbnoise chapter for noise information.

Setting Harmonics and moderate Sidebands

In the qpss analysis, when shooting is selected, each slice of the Large signal (usually the LO signal) has a minimum of 200 timepoints, so it inherently has frequency domain content through the 100th harmonic of the signal in each slice. Only the harmonics specified for the moderate tones affect the accuracy of the qpss solution. If you specify enough harmonics in the moderate tones in the qpss analysis to accurately capture the circuit performance, then the qpsp solution will be accurate.

To see if you have enough moderate tones, set a number of moderate harmonics, and run the simulation. Make the measurement. Now increase the number of moderate harmonics by about 50%. Run again, and make the same measurement. If the measurement did not change significantly, you had enough moderate harmonics to begin with, and you might be able to reduce the number of moderate harmonics. If it did change, then you need more moderate harmonics. If your moderate signal is below compression, start with two or three harmonics. If the moderate signal is at compression, start with five harmonics. If the moderate signal is above compression, start with about seven harmonics.

© 1994-2020 All Rights Reserved.

ADE Implementation

This section jumps from theoretical to the practical with examples of how to use qpsp. The focus is only on the qpsp *Choosing Analyses* form so you can see where the individual settings are made. To see examples with all the steps shown, see the examples section that follows this section.

QPSP Setup

For this example, consider an LNA-Mixer circuit shown below.

© 1994-2020
All Rights Reserved.

1. First, define either an input or output frequency range in the *Frequency Sweep Range* fields. It is usually easier to define an input frequency range, as shown below for a 2.4GHz application.

2. Linear and log sweeps are provided. It is usually better to select Linear or Log, and specify the number of steps instead of using automatic. Automatic will run 50 frequency points, which usually is not needed for most applications.

© 1994-2020
All Rights Reserved.

3. Now port and frequency definitions need to be made.
4. Type 1 in the Port # column field. Click *Select Port*, and select the input port.

© 1994-2020
All Rights Reserved.

5. Click *Choose Harmonic*. In the window that appears, select the input frequency range. Now select *OK*.

© 1994-2020
All Rights Reserved.

6. Click Add. The port-frequency combination is added to the list. By convention, port 1 sets the input frequency for qpsp.

© 1994-2020
All Rights Reserved.

7. Now define the output port as port 2.

© 1994-2020
All Rights Reserved.

8. Select the output frequency range by clicking *Choose Harmonic*, and selecting the output frequency range as shown below.

© 1994-2020 All Rights Reserved.

9. Next, click *Add*. This adds the port/frequency combination as port 2 in the list. By convention, port 2 sets the output frequency.

© 1994-2020
All Rights Reserved.

10. If necessary, repeat the process to define port 3 as the gain from RF to IF with no frequency translation. (The RF to IF isolation.)

© 1994-2020
All Rights Reserved.

11. If you want a noise simulation, define the *Maximum Sideband* term above. For more information, see the pnoise section in this chapter.

© 1994-2020
All Rights Reserved.

QPSP Options

Below is the qpsp options form.

None of the options are commonly used.

tolerance

Leave this option at the default value.

Qpsp uses an iterative solver to calculate the S-Parameter measurements. Any iterative solver needs an error tolerance to specify when to stop iterating because the solution is accurate enough. The tolerance option specifies that accuracy for qpsp. For shooting, the default tolerance is 1e-9. For circuits where HB is the qpss engine, the default is 1e-6.

© 1994-2020 All Rights Reserved.

gear_order

Do not change this option.

solver

The default solver is the turbo solver.

When hb is used as the engine in qpss, leave this option at the default.

When shooting is used, sometimes when the qpsp analysis frequency is very close to the frequency of one of the harmonics in the qpss analysis, warning messages will appear in the qpsp output warning that the accuracy might not be good enough. If you see these messages, select the *std* solver, which has better ability to handle frequencies that are very close to a harmonic in the qpss, but which takes longer to run than the turbo solver.

resgmrescycle

Leave this option at the default. For the resgmres linear solver, there are several different options.

hbprecond_solver

This option is available only when harmonic balance selected in qpss and APS is used.

The basic solver is the only solver available in standard Spectre when harmonic balance is selected. Autoset is the default solver in harmonic balance when APS is used. This solver is faster, but occasionally stagnates. When stagnation is detected, APS automatically switches to the basic solver, and prints a message in the Spectre output window. If you have stagnation, it will save a small amount of time to set this option to basicsolver.

annotate

This option controls the level of detail in the Spectre output log. No detail is provided when you select no, while more details are provided as you move towards right with steps option providing maximum detail.

© 1994-2020 All Rights Reserved.

freqaxis

Qpsp displays all the results on the same frequency scale. The default is absin (absolute value of the input frequency). If there is a negative input frequency, in will show that frequency on the positive frequency axis. The selection In also shows the input frequency, but if the input frequency is negative, it will be shown on a negative frequency axis. The selection Out shows the output frequency range. If the output frequency is negative, it will show on a negative frequency axis.

file

Outputs S-parameter data into the specified file.

datafmt

Specifies the data format of the S-parameter output file. Possible values are *spectre*, *touchstone*, and *touchstone2*. The default value is *spectre*.

datatype

Specifies the data type of the S-parameter output file. Possible values are *realimag*, *magphase*, and *dbphase*. The default value is *realimag*.

additionalParams

additionalParams is typically used for new features that are being beta tested. Keyword=value pairs are expected in this field.

For more information about the other options, type `spectre -h qpsp` at the command prompt in a Unix shell window.

© 1994-2020 All Rights Reserved.

Examples

For this example, consider an LNA-Mixer circuit shown below.

© 1994-2020
All Rights Reserved.

1. First, define either an input or output frequency range in the frequency sweep range fields. It is usually easier to define an input frequency range, as shown below for a 2.4GHz application.

2. Linear and log sweeps are provided. It is usually better to select Linear or Log, and specify the number of steps instead of using automatic. Automatic will run 50 frequency points, which usually is not needed for most applications.

© 1994-2020
All Rights Reserved.

3. Now port and frequency definitions need to be made.
4. Type 1 in the *Port #* column field. Click *Select Port*, and select the input port.

© 1994-2020
All Rights Reserved.

5. Click *Choose Harmonic*. In the window that appears, select the input frequency range. Next, click *OK*.

© 1994-2020
All Rights Reserved.

6. Click *Add*. The port-frequency combination is added to the list. By convention, port 1 sets the input frequency for qpsp.

© 1994-2020
All Rights Reserved.

7. Now define the output port as port 2.

© 1994-2020
All Rights Reserved.

8. Select the output frequency range by clicking *Choose Harmonic*, and selecting the output frequency range as shown below.

© 1994-2020
All Rights Reserved.

9. Next, click *Add*. This adds the port/frequency combination as port 2 in the list. By convention, port 2 sets the output frequency.

© 1994-2020
All Rights Reserved.

10. Repeat the process to define port 3 as the gain from RF to IF with no frequency translation. (The RF to IF isolation.)

© 1994-2020
All Rights Reserved.

11. If you want a noise simulation, define the maximum sideband term above. For more information, see the pnoise section in this chapter.

© 1994-2020
All Rights Reserved.

12. Now run the simulation. When the simulation completes, select *Results - Direct Plot - Main form* in the ADE Explorer window. The *Direct Plot Form* is displayed.

13. Note that only the S-Parameters and the noise calculations are available for plotting in the *Direct Plot Form*.

14. Select *Z-Smith* in the *Plot Type* section in the *Direct Plot Form*. Now plot S11 on the Smith Chart.

© 1994-2020 All Rights Reserved.

15. Z-Smith is shown for the S_{11} plot below. The marker is placed at 2.4GHz.

16. In the waveform tool, there are two zoom features that are helpful. Typing the f key fits the data to the screen. It will not necessarily display the unit circle, depending on the actual S-Parameters to be plotted. <Shift>-f fits the Smith Chart to the unit circle as shown above. This is useful for getting the big picture of the plot.

© 1994-2020 All Rights Reserved.

17. When S_{21} is plotted, the curve goes outside the unit circle. The small green trace in the center is the S_{11} plot from the previous graphic.

© 1994-2020
All Rights Reserved.

18. Rectangular plots are also available. Select *Rectangular* in the *Plot Type* section in the *Direct Plot Form*, as shown below.

© 1994-2020
All Rights Reserved.

6

Envelope (ENVLP) Analysis

Envelope analysis is used instead of a transient analysis when there are fast things and slow things in the same circuit. The simulation term for this condition is called multi-rate simulation. In the transient, a large number of cycles of the fast signal must be simulated in order to capture the slow behavior in the circuit. Instead of simulating every fast cycle as in the transient, envelope skips many cycles of the fast signal because the change is very slow. Although many cycles are skipped, the result is almost as accurate as transient analysis.

The primary application for envelope analysis is for digital modulation of a high frequency carrier, or calculating the startup waveform of a high-Q oscillator.

Envelope can solve for one cycle of the RF carrier either by solving for the waveform when shooting is selected, or by using harmonic balance when harmonic balance is selected. For very nonlinear systems like polar modulation, use shooting. For relatively more linear circuits like and I/Q modulator and or power amplifier, use harmonic balance. Harmonic balance has the advantage of having two fast envelope modes that can dramatically speed up simulation.

Harmonic balance envelope has two different algorithms. The first is to simulate everything at the transistor level. The second, Level1, characterizes the circuit at the start of the simulation, and builds a behavioral model that is then run instead of the circuit. Level1 is useful for circuits that do not have memory effects. A level 2 choice is present in the ADE Explorer *Choosing Analyses* form, but when this is selected level 1 is actually used. You can see a message in CIW when the simulation starts.

Example

Consider a circuit with a wireless signal source, a behavioral PA, a wireless probe, and a 50 ohm resistor, as shown below. Multiple wireless probes are allowed. Connect a wireless

© 1994-2020 All Rights Reserved.

probe at each point in the circuit you want to measure. The wireless source can be at any level in the hierarchy.

© 1994-2020
All Rights Reserved.

The wireless source from `rfLib` generates a fully compliant LTE signal.

© 1994-2020
All Rights Reserved.

The behavioral PA has 20dB of gain, and is slightly nonlinear.

Now start ADE Explorer, and open the envelope *Choosing Analyses* form. The form is blank.

© 1994-2020
All Rights Reserved.

Next, select the wireless mode, and type in the number of harmonics you want to simulate. Generally, `level1` fast envelope is used because of the speed advantage compared to transistor-level simulation. Both modes work with wireless.

© 1994-2020
All Rights Reserved.

Now run the simulation from the ADE Explorer window. When you start the simulation for the first time, a variable `W0_wfreq` is introduced into the ADE variables list. This variable is used only when modulators are present in the design, and this variable should be used to set the LO frequency in the modulator. The actual frequency used in simulation is defined by the frequency range and channel number that is defined in the RF source.

© 1994-2020
All Rights Reserved.

Now open the *Direct Plot Form*. All the common measurements are available from the *Direct Plot Form*. For all wireless plots, select the wireless analysis choice at the top of the form. To plot the spectrum, select *Spectrum*, and then select the probe you want to plot.

© 1994-2020
All Rights Reserved.

Click *Plot*, and the spectrum is plotted. The spectral mask for the appropriate standard is also plotted. The channel power and the high and low acpr measurements are displayed using markers.

© 1994-2020
All Rights Reserved.

To plot the high and low acpr measurements at higher frequency offsets, select *Customized ACPR* in the *Direct Plot Form* and specify the number of channels in the *Number of Adjacent Channels* field, as shown below.

© 1994-2020
All Rights Reserved.

Click *Plot*. The high and low acpr measurements at higher frequency offsets are displayed using markers.

© 1994-2020
All Rights Reserved.

To plot the constellation diagram, select *Constellation*, and then either the *Percent* or dB value for the EVM measurement. Now select either the measured constellation, or the ideal constellation fitted to the actual datapoints, and click *Plot*.

© 1994-2020
All Rights Reserved.

The constellation is displayed along with the EVM calculation. For the 802.11n standard, EVM for the data, pilot, or the weighted average of both is available.

© 1994-2020
All Rights Reserved.

To plot the subframe by subframe EVM, select *Measure*, and then select the <*probe name*> *.evm*.

© 1994-2020
All Rights Reserved.

The EVM subframe by subframe is displayed.

Loadpull

For a loadpull simulation, use a port for the load.

1. Select the *Loadpull* checkbox.

2. Click the *Select* button on the right of the grayed out *Load Instance* field.

© 1994-2020
All Rights Reserved.

3. In the schematic, select the load port.

   The instance name is entered in the *Load Instance* field. Direct entry by typing is not allowed for this field.

4. Click the *Variable* button on the right of the *rho (Z)* field.

5. In the *Select Design Variable* window, select the variable you entered in ADE Explorer.

   This variable will be used to set the value of the reflection coefficient in the port.

6. Specify a sweep range and set the spacing.

7. Click *OK*.

8. Click the *Variable* button on the right of the *phi (Z)* field. This sets the angle of the reflection coefficient.

9. In the *Select Design Variable* window, select the second variable you entered and specify the sweep limits.

10. Click *OK*.

    The *Choosing Analyses* form is updated.

11. Specify the system resistance in the *Z0* field.

© 1994-2020 All Rights Reserved.

The *Choosing Analyses* form should appear similar to the one below.

12. When completed, click *OK* and run the simulation in ADE Explorer.

© 1994-2020
All Rights Reserved.

13. In ADE Explorer, select *Results - Direct Plot - Main Form*.

The Direct Plot Form is displayed as shown below.

14. Click *Plot* with the settings as shown in the image above.

A smith chart is plotted.

© 1994-2020
All Rights Reserved.

© 1994-2020
All Rights Reserved.

CCDF Post Processing

You can perform complementary cumulative distribution post processing (CCDF) for each probe placed in the circuit. Consider the following example design:

To plot CCDF:

1. Run a fast envlp wireless simulation.
2. Open the Direct Plot Form.

   CCDF will be listed in the Function group box if there is an `envlp.ccdf.envlp` file in your `.raw` directory.

© 1994-2020 All Rights Reserved.

3. Select the *CCDF* function.

Selected Probe (sig) to plot lists the wireless probes relative to CCDF. `.ccdf`, `.avgPower`, and `.peakPower` probe types are included in this list.

4. Select the required probe and click *Plot* or *Add To Outputs*.

The plot results will be based on the type of simulation that was run and the probe type that is selected.

For non-sweep simulations

© 1994-2020 All Rights Reserved.

If a non-sweep simulation is run and the `.ccdf` probe type is selected in the Direct Plot Form, a curve will be plotted as shown below:

If a non-sweep simulation is run and the `.avgPower` or `.peakPower` probe type is selected in the Direct Plot Form, there will be no option to plot the results because they are scalar values. Select the *Add to Outputs* checkbox to add these probes to the outputs as shown below:

After adding the expressions to *Outputs Setup* tab, open the *Results* tab and click the *Re-evaluate* button to update the values in the results table.

Note: For updating the results table, rerunning the simulation is not requried.

© 1994-2020
All Rights Reserved.

For sweep simulations

Consider a simulation is run with the following sweep settings:

```
snr1=10,snr2=20;temp1=27,temp2=50
```

For this simulation, you can plot results for all the probe types using the Direct Plot Form.

Example of CCDF plot:

© 1994-2020
All Rights Reserved.

Example of average power plot:

Example of peak power plot:

© 1994-2020
All Rights Reserved.

Envelope Principles

Harmonic Balance

Use harmonic balance if the circuit has sinusoidal waveforms and is fairly linear. This is true for circuits with sinusoidal waveforms operating below compression. If polar modulation is used where the circuit is operated in a very nonlinear mode, then shooting is suggested.

Basic Principles

Consider an amplifier that is being driven by an unmodulated source. Because of the finite bandwidth of the system, there is some startup behavior, and then the amplifier reaches steady-state.

The time-zero time point is the initial transient solution, which is a DC analysis that observes all the time-zero values for all the waveforms in the system.

The waveform above is an actual starting waveform for harmonic balance. Vertical markers have been placed at 400nsec intervals, which correspond to one cycle of 2.5GHz.

© 1994-2020
All Rights Reserved.

When the system becomes predictable with a low order polynomial, harmonic balance envelope begins to skip cycles. The number of cycles it skips depends on whether fixed or adaptive timestep control is selected. For fixed timestep control, the next strobing point is simulated. For adaptive timestep control, the number of cycles skipped depends on the curvature of the envelope. For an unmodulated sinusoid, conceptually, the cycle skipping could continue to grow to the end of the simulation because the same waveform repeats over and over again.

Note that for an accurate PSD and ACPR measurement, the data needs to be strobed at the time interval in the original I and Q modulation file. In harmonic balance, there does not need to be an integer number of skipped cycles.

If that delta-time was 8.3333333333nsec, to get an integer number of skipped cycles, the carrier frequency needs to be an integer multiple of 120MHz. Note that when the carrier is 2.4GHz, this is a multiple of 20. Because of the integer multiple, the envelope time-domain waveform shows a horizontal line between the cycles of simulation.

© 1994-2020
All Rights Reserved.

When the carrier is 2.41GHz, there is a small offset from the actual period of the carrier frequency. This causes a sinusoidal line to appear between the cycles that are simulated. This is because the periods do not exactly align.

Note that harmonic balance envelope is not a series of steady-state harmonic balance simulations. It is a hybrid of transient and harmonic balance.

© 1994-2020
All Rights Reserved.

Consider an amplifier followed by a diode peak detector shown below.

© 1994-2020
All Rights Reserved.

At time zero, there is no signal, so the filtered output at the net *out2* starts at zero. The resistors and capacitors cause the signal to ramp up in an exponential fashion.

If the envelope simulation had been steady-state at each time point, the voltage at the net out2 would have been constant and non-zero from the start.

For systems that can have complex behavior, make sure that adaptive time step control is selected. This allows the number of skipped cycles to be dynamic along with the variations in the envelope signal. If this system had been an analog AGC circuit, the feedback might have resulted in overshooting and ringing. This behavior could accurately be determined by using adaptive timestep control.

Envelope-Transient Simulation

In envelope-transient analysis, a circuit is driven by one or more modulated carriers of the form:

$$x(t) = Re\left\{X(t)e^{jw_o t}\right\} = Re\left\{A(t)e^{j\theta(t)}e^{jw_o t}\right\} = Re\left\{[I(t) + jQ(t)]e^{jw_o t}\right\}$$

© 1994-2020 All Rights Reserved.

Subject to the modulated carrier excitation, circuit waveforms can be represented by sums of slowly modulated carriers harmonically related to w_0. Every voltage, charge, and current looks like:

$$v(t) = \sum_{n=-N}^{N} V(t)e^{jnw_0 t}$$

Envelope-transient analysis solves for the harmonic modulations (complex envelopes) $V_n(t)$ at each node. Because the envelope bandwidths are typically much lower than the carrier frequency, the sampling rates are correspondingly lower and the simulations more efficient than traditional transient analysis.

Under modulated carrier excitation, circuit equations take the form:

$$\frac{d}{dt}Q(V(t)) + jwQ(V(t)) + F(V(t)) = X(t)$$

where $V(t)$ is the vector of nodal harmonic envelopes. The remaining terms represent the quasi-static contributions to the envelope circuit equations, equivalent to a repeated solution of steady-state circuit equations by means of harmonic balance, one for each envelope time step t_i, where the sources are treated as sinusoids having a time-varying complex amplitude $X(t_i)$.

The dynamic term describes the contribution of reactive elements to the envelope circuit equations. Taking a linear capacitor as an example, it becomes:

$$C\frac{dV(t)}{dt}$$

This term vanishes unless the modulation is relatively wideband and/or the capacitance is relatively large.

Transistor Level V/s Fast Envelope

Transistor-level (envelope-transient) simulation solves for all of the circuit equations rigorously, taking dynamic effects into account. The cost of transistor-level envelope

© 1994-2020 All Rights Reserved.

simulation is virtually the same as a sweep of M harmonic balance simulations, where M is the number of envelope time samples. Transistor-level envelope analysis is usually more efficient than transient simulation for modulated inputs, but it remains computationally very expensive when the number of the time samples is large, as is usually the case for today's communication standards.

The fast envelope method relies on two basic assumptions to accelerate simulation speed:

- The modulation is relatively slow and the contribution of the dynamic term mentioned earlier is small or negligible.
- There is a single-modulated carrier at the input (or, equivalently, a pair of baseband I/Q channels).

With those assumptions and considering the output signal as a modulated carrier of the form,

$$y(t) = Re\left\{ Y(t)e^{jw0t} \right\}$$

the output envelope is a function of the input signal's amplitude and phase:

$$y(t) = f(A(t), \theta(t))$$

Two forms of f(A(t) , θ(t)) are observed in practice:

The Passband Model

In the passband model the nonlinear behavior is dependent on the magnitude of the input signal only.

$$f(A,\theta) = G(A)e^{j\theta}$$

`G(A)` describes the circuit's amplitude and phase compression characteristic. It is extracted efficiently by treating the input signal as a sinusoid and sweeping its amplitude over the expected range of the complex envelope's magnitude. At runtime, the output is obtained by table look-up and interpolation, which are very efficient operations. Since only a handful of points are required to build the look-up table, typically around 10, the runtime cost of the passband fast envelope model is equivalent to approximately 10 harmonic balance simulations. For a typical modulation source consisting of thousands, or tens of thousands of

© 1994-2020 All Rights Reserved.

samples, the speed-up offered by fast envelope is much higher than the transistor-level methods.

The Baseband Model

The most frequent application of the passband model is in PA design where the input,

$$x(t) = Re\left\{X(t)e^{jw_o t}\right\} = Re\left\{A(t)e^{j\theta(t)}e^{jw_o t}\right\} = Re\left\{[I(t) + jQ(t)]e^{jw_o t}\right\}$$

may be viewed as a pair of I/Q channels upconverted by an ideal quadrature modulator. For most other situations, especially when the circuit contains transistor-level modulators, the output is a function of both the I and Q channels (or, equivalently, the modulated carrier's amplitude and phase), and therefore, requires a 2D sweep for proper characterization.

SpectreRF now includes a highly efficient and accurate baseband fast envelope model, which typically requires approximately 100 or fewer harmonic balance sweep points for proper characterization.

Fast envelope models are extracted from swept harmonic balance simulations performed over a range of the input envelope magnitude (for the passband model) and phase (for the baseband model). The number of sweep points has a significant impact on both the simulation accuracy and on speed of model extraction. In the present implementation, by default, the number of sweep points is set at 7 for the magnitude and (in baseband model only) 12 for the phase. This is a controllable parameter set by the options sweepmethod, Magnitude Points, and Phase Points in the envelope options form. Choosing Fine for sweepmethod will run 10 magnitude points and 12 phase points. Setting *userdefined* allows you to set both the number of magnitude and phase points for the characterization phase of the fast envelope extraction. Adaptive is a work in progress.

Power Scaling in Baseband Mode

In passband mode, the power specified by the `wsource` instance parameter *Power* represents the average power dissipated in a matched load. In baseband mode, *Power* represents the sum of the average power dissipated in matched load resistors in the I and Q channels. More precisely, assuming that the power available in the I and Q channels is the same, the sum of power in the I/Q loads is:

© 1994-2020 All Rights Reserved.

$$P = \frac{2}{RT}\int_0^T v^2(t)dt$$

Here, `v(t)` is the voltage across either of the load resistors. Then,

$$v_{RMS} = \sqrt{RP/2}$$

Using `P=10 mW` and `R=50` as examples, the rms voltage should be 500 mV.

WFreq

Notice that the fundamental frequency is specified as `W0_wfreq`. The variable, termed `wfreq`, is entered on your behalf directly from the wsource instance name. In general, its format is `<wsource instance name>_wfreq`.

`wfreq` is a reserved variable, which represents the carrier frequency of the source. Each standard has channel numbers and band numbers (if applicable) that define the carrier frequency that is currently in use.

Instead of calculating carrier frequencies manually, you can directly access the frequency computed by the source by means of `wfreq`.

`wfreq` is available in the analysis form but you can also use it to set values of the instance parameters.

When you click *OK*, ADE Explorer will automatically add a variable called `<wsource instance name>_wfreq` to the list of *Design Variables* and set it to `1G`. The wireless engine overrides it with the correct value internally and issues an info message in the log to report the actual value used during simulation. The actual value of the carrier frequency is reported in the Spectre output window.

When do I use Passband or Baseband?

Use the following simple rules to choose the model:

© 1994-2020 All Rights Reserved.

- If the input is a pair of baseband I/Q signals upconverted by a transistor-level (or otherwise non-ideal) modulator, use the baseband model.
- If the input is a modulated carrier (typically a case for PA modeling), use the passband model. An uncommon exception to this rule is the modeling of demodulators. Since the output of a demodulator depends both on the real and imaginary parts of the input signal, use the baseband method to model it.

Fast Envelope Noise

Optionally, in the presence of random noise, the output is modeled as:

$$y(t) = Re\left\{[Y(t) + N(t)]e^{jw0t}\right\}$$

The random noise signal, `N(t)`, is extracted by one of the two methods: linear noise analysis around the DC operating point, or periodic noise analysis computed around a sinusoidal input (or the equivalent DC representation in the case of baseband input) given by:

$$\sqrt{\overline{A^2(t)}}e^{j\overline{\theta(t)}}$$

In both cases, the noise analysis is performed over the modulation bandwidth to extract the noise spectrum and calculate its complex-envelope representation. The periodic noise representation takes more time to extract than the linear model but is recommended as more rigorous and accurate than the linear approximation.

To perform envelope noise simulation:

© 1994-2020 All Rights Reserved.

1. Open the Envelope Choosing Analyses form, as shown below.

© 1994-2020
All Rights Reserved.

2. Select the *Wireless* option.
3. Specify the number of harmonics that you want to simulate in the *Number of Harmonics* field.
4. Select *level1* for *Fast envlp mode*.

   Here, *level1* refers to linear noise and *level2* refers to pnoise, as explained above.
5. Select the *Noise* check box.
6. Leave the *Sweep type* to the default value of *Log*.
7. Leave the *Noise method* to the default value of *level2*.
8. Set the start frequency at least 4 decades smaller than the bandwidth of the modulation but not equal to zero.

   The stop frequency is set automatically based on the bandwidth of the modulation.
9. Click *Apply*.

Wireless Mode in Envelope Analysis

Starting with the 13.1 release of SpectreRF, RFIC designers can evaluate system-level performance metrics in an integrated, automated, and easy-to-use simulation flow enabled by the new wireless analysis option.

In wireless analysis, standards-compliant modulation sources are applied to the DUT and measurement probes calculate system-level performance metrics. The sources and the

© 1994-2020 All Rights Reserved.

probes are implemented by two new Spectre primitives: `wsource` and `wprobe`. Based on the familiar SpectreRF envelope-following algorithms, the wireless engine works with `wsource` and `wprobe` to automate the analysis setup and simulation parameters. The sources and the probes are standard-aware, resulting in a powerful and streamlined flow - though rich in standard support and complete in system-level simulation coverage, it is designed from the ground up for everyday use by RFIC designers.

Note: Wireless sources are only valid in wireless analysis. They cannot be used in traditional envelope analysis.

Supported Standards

The 19.1 release supports Bluetooth, DMR, LTE, NR, ZigBee, Smart Meter (802.15.4g),802.11a, 802.11b, 802.11n, 802.11p, 802.11ac, 802.11ad, 802.11ah, 802.11ax, and 802.11af standards.

© 1994-2020 All Rights Reserved.

Wireless Analysis Vs. Traditional Envelope Following

Wireless analysis, based on traditional envelope-following analysis, has the advantage of automated set-up and post processing using the new `wsource` and `wprobe` components. Advanced modeling options, such as fast envelope, are supported as well. Wireless analysis produces, among other features, the traditional envlp dataset, making it compatible with all of the pre-existing Direct Plot capabilities and post-processing functionality.

Wireless analysis offers a number of additional benefits beyond the traditional envelope following:

- **Parametric modulation sources**. Traditional envelope-following relies on generic PWL sources of fixed frame structure, sampling rate, bit patterns, and modulation options. In contrast, wsources are fully parametrized according to the specified communications standard.
- **Automated simulation setup and control**. Wireless analysis works with standards-aware wsources, which help determine all of the relevant simulation parameters – sampling rates, stop times, strobe options, and carrier frequencies, as dictated by the communications standard. A key benefit of automated simulation control is that the designer is now less likely to make common setup mistakes which affect simulation and measurement accuracy. Wireless analysis is fully supported within ADE Explorer behind an easy-to-use and familiar-looking analysis setup form.
- **Efficient post-processing**. Most of the wireless analysis post processing is performed at runtime by wprobes. Much like the sources, wprobes are aware of the communications standard specifications. They output all of the standard-specific system-level metrics, such as EVM, BER, and spectrum masks directly to the output database. System-level metrics are available directly from the Direct Plot form, without manual post processing and in compliance with the modulation format and communications standard.

Limitation of Wireless Analysis

Wireless analysis works in conjunction with Cadence-supplied `wsources` and `wprobes`. Custom PWL sources are not supported in this mode. Traditional envelope-following can still be used with custom PWL sources as a fully supported simulation option.

Many wireless simulations are done using level 1 fast envelope. Level 1 fast envelope allows noise to be simulated but does not take memory effects into account. Although you can use wireless simulation (without level 1 fast envelope) for circuits with memory effects, it is recommended that you use transistor-level envelope simulation for such circuits.

© 1994-2020 All Rights Reserved.

Shooting

Wireless mode is not available when shooting is selected.

Shooting is suggested if polar modulation is used where the circuit is operated in a nonlinear mode. Use harmonic balance if the circuit has sinusoidal waveforms and is fairly linear. This is true for circuits with sinusoidal waveforms operating below compression.

Consider an amplifier that is being driven by an unmodulated source. Because of the finite bandwidth of the system, there is some startup behavior, and then the amplifier reaches steady-state.

The time-zero time point is the initial transient solution, which is a DC analysis that observes all the time-zero values for all the waveforms in the system. At this point, the only thing that is known is the time-zero value, so the projected ending point of the first period is the starting point.

The waveform above is an actual starting waveform for shooting. Vertical markers have been placed at 400nsec intervals, which correspond to one cycle of 2.5GHz.

Note that at the end of the first period, the ending point misses the starting point by a lot. This error between the predicted and actual ending point is used to control the simulation. In this case, the error is large, so the transient continues for another period. The beginning and

© 1994-2020 All Rights Reserved.

ending points of the first cycle allow a prediction of a ramp as the ending point for the second cycle.

When the system becomes predictable with a low order polynomial, shooting envelope begins to skip cycles. First, it will skip one period, which is done twice in a row in the above example. Then it skips 2 periods, and then it skips 4 periods. For an unmodulated sinusoid, conceptually, the cycle doubling could continue to the end of the simulation because the same waveform repeats over and over again.

Shooting has the limitation that only an integer number of cycles can be skipped. This is because the starting and ending voltages need to be predictable, which happens only when an integer number of cycles is skipped.

When digital modulation is present, the number of skipped cycles depends on the curvature of the envelope. When the envelope has little curvature, many cycles can be skipped. When the envelope has a large curvature, not many cycles can be skipped.

In order to maintain accuracy, adaptive timestep needs to be used with shooting to allow the envelope algorithm to maintain the accuracy of the simulation.

Note that for an accurate PSD and ACPR measurement, the data needs to be strobed at the time interval in the original I and Q modulation file. For maximum accuracy, the carrier frequency needs to be an integer multiple of the reciprocal of the delta-T in the I and Q modulation file.

© 1994-2020 All Rights Reserved.

If that delta-time was 8.3333333333nsec, the carrier needs to be an integer multiple of 120MHz. Note that when the carrier is 2.4GHz, this is a multiple of 20. The sampled carrier cycles are exactly aligned with the data in the I and Q modulation file.

© 1994-2020
All Rights Reserved.

When the carrier is 2.45GHz, because of the limitation of skipping an integer number of cycles, the timing of the actual points in the envelope is not an integer multiple of 8.3333nsec. This causes small interpolation errors in the result due to the time mismatch.

Shooting envelope is a general-purpose extension of the transient and is useful for calculating the startup behavior of many circuits that have very long time constants and very short time constants in the same circuit. An example is analog AGC (Automatic Gain Control).

© 1994-2020 All Rights Reserved.

As an example, consider an amplifier that has a diode peak detector after the amplifier.

© 1994-2020
All Rights Reserved.

The capacitor voltage starts at zero because there is no signal at time zero, and then has an exponential charging curve set by the resistors and capacitor in the peak detector.

ADE Implementation and Detailed Envelope Settings for a Driven Circuit

First a modulation source is needed. Your choice of modulation source depends on whether you will be using wireless mode or using your own I and Q modulation files. Cadence provides three components in *rfLib* called *wlan11n, lte, or ZigBee*, which have modulated signals for the three standards. If you use one of these sources, then add wprobes in your circuit to make the measurements.

If you are using your own I and Q modulation files, you will need to get a piecewise linear I and Q modulation file from your system design group. These files will have a constant delta-time between the points. This delta-time should be 1/(channel bandwidth*6) or smaller if you need an ACPR measurement, and 1/(channel bandwidth*10) or smaller if you need a second adjacent channel power ratio calculation. This is required to meet the Nyquist criteria for the modulation.

To create the modulated RF signal from the I and Q modulation files, either a port component from *analogLib* or *ACPR_Source* from *rfLib* can be used. If the port is used, you need to scale the amplitude of the RF signal to achieve the correct power at the output of the port. If you use the *rfVsource*, it will read the I and Q data files, and the power in dBm can be set

© 1994-2020 All Rights Reserved.

directly on the source. Note that this power will be achieved when you simulate to exactly the end time in the I and Q modulation files. The properties list for the port is shown below.

© 1994-2020
All Rights Reserved.

The properties list for the ACPR source is shown below.

The raised cosine filter should not be needed for most simulations.

Setting up the Envelope Analysis in ADE Explorer: Wireless Mode

If you are using wireless mode, all you have to do is place a wireless source in the schematic, add wireless probes in the schematic at every point you want to make a measurement, select wireless mode in ADE Explorer, and define a reasonable number of harmonics for your system. After you run the envelope analysis, direct plot functions are available for all the common measurements. See the example at the beginning of this chapter for details.

Setting up the Envelope Analysis in ADE Explorer: Using your own I and Q Files

1. Everything starts with the I and Q files you are working with.

© 1994-2020 All Rights Reserved.

2. Calculate 1/time difference in these files. If you are using shooting, consider making the carrier frequency in the design an integer multiple of this frequency. This will reduce the numerical noise floor of the simulation. For harmonic balance, it is not needed unless you are trying to determine the phase of the signal based on the time-domain result. The carrier frequency has been set in ADE Explorer variable *frf* shown below.

3. Now select *Analysis - Choose* or click the *Choosing Analyses* icon on the right side of the ADE Explorer window.

© 1994-2020 All Rights Reserved.

4. Select *envlp*. The ENVLP *Choosing Analyses* form is displayed.

5. Now click the *Start ACPR Wizard* button.

© 1994-2020
All Rights Reserved.

The ACPR wizard is displayed.

© 1994-2020
All Rights Reserved.

6. From the *Clock Name* drop-down list, select the name that is in the *Frequency name* 1 or 2 property that sets the carrier frequency in the circuit. If you are using *ACPR_source*, this name is *fc*.
7. Select *Net* or *Differential net*, depending on your design.
8. Click *Select* on the right of the *Net* field and then select the output net(s) in the schematic.
9. If you are working on a Qualcomm standard, select *Power Density*. Otherwise, select *Power*.
10. If you are working on an 802.11n standard, select the standard from the Channel Definitions drop-down list. The rest of the fields in the form will have starting values. Please see the discussion for Symbol Start, Window Size, and Repetitions below.
11. If you are working on another standard, select Custom.
12. Type the channel bandwidth in the *Main Channel Width* field.
13. In the Adjacent frequencies section select *low*.

    The edit fields below this section become active.
14. Type the frequency limits for the first low adjacent channel for the ACPR calculation. Note that these frequencies are referenced to the center of the main channel. For frequencies on the low side of the main channel, the numbers will be negative.
15. Select *Change*.

    This moves the contents of the edit field up into the list.
16. In a similar manner, select *high*, enter the frequency range, and click *Change*.
17. If you have second adjacent specifications, add the name and frequency limits in the edit field above, and then click *Add*.
18. Set *Symbol Start(Sec)* field to the time when you want the Fourier transform to begin. In many I and Q files, there is a time delay caused by digital filtering. The start time for the frequency response calculation needs to be after that delay time. If you are unsure,

© 1994-2020 All Rights Reserved.

simulate the I and Q file in a piecewise-linear source in the transient analysis, and watch for the time that the output becomes non-zero, and pick a time slightly later than that.

19. Set the strobe period to the delta-time in the I and Q modulation file. If you are unsure, view the file in a text editor, and look at the left entry of the second line. The first line should have a left entry of zero (seconds) The second entry left entry (in seconds) should be entered in the strobe period field. The right entry is the voltage at the time in the left column.

Note: When you enter a strobe period, the ACPR wizard will update the Symbol Start field so that it is an integer multiple of the strobe period.

The Fourier transform used to calculate the frequency response of the circuit is inherently noisy. To make the spectrum less noisy, the *psd* function is used. The *psd* function splits the time that is simulated into smaller pieces, which allows multiple shorter time sequences to be considered individually. A fourier transform is performed on all the pieces individually, and then the power of all the first harmonics is averaged. This is done

© 1994-2020 All Rights Reserved.

for the rest of the harmonics also. Because of the averaging, the noise is reduced. Because a shorter time exists in each piece, the frequency of the first harmonic goes up. In the *psd*, this reduces the resolution (the number of frequency points analyzed in the spectrum). Generally, a trade-off is needed between the number of pieces, the length of the simulation, and the noise in the psd calculation.

20. Window size is the number of delta-times in the I and Q input files you want to include for the spectral calculation. This needs to be a power of two. Generally, this is in the range of 256 to 8192.

21. Specify repetitions as a power of 2. This will run as many windows sizes as you specify.

22. The *Symbol Stop(Sec)* is calculated based on the symbol start, window size, and repetitions fields. Make sure that the data in your I and Q modulation file has entries to at least this time. View the I and Q modulation file, and look at the left entry (the time) in the last line of the file to make sure that ending time is larger than the *Symbol Stop(Sec)* time in the ACPR wizard.

The resolution bandwidth is calculated based on the entries for strobe period and window size. It is the frequency resolution of the Fourier transform of the output signal.

23. Because the data bits in the modulation are random, a Fourier transform window function needs to be selected. There are advantages and disadvantages to all window functions. The *Hanning* window is a reasonable window if the window function is not specified by the standard. Check with your system designer to find out his or her preference.

24. When you are done, click *OK*.

In the *Choosing Analyses* form for ENVLP, the *Stop Time* field is populated. The *Time Step Contro*l is also set to *fixed* and the *Step Period* field has the entry that you specified for the strobing interval in the ACPR wizard. A number of envelope options are

© 1994-2020 All Rights Reserved.

also set to provide the output from the simulator at the times needed for the FFT-based frequency response calculation.

© 1994-2020
All Rights Reserved.

25. Either type the frequency of the carrier, the number of harms, and the oversample factor, or select names, and set the desired number of harmonics.

26. If you have square waves in your circuit, use an oversample factor if you choose harmonic balance, or select shooting. If you decide to use harmonic balance, you need relatively more harmonics. In this case, also set oversample factor to two to eight. The more non-sinusoidal the waveform is, the higher oversample factor should be.

© 1994-2020 All Rights Reserved.

27. If you have a frequency divider in your circuit, specify the divider ratio in the *Freqdivide Ratio* field.

28. For the *Time Step Control* field, select *fixed* or *adaptive* for Harmonic Balance and *adaptive* for Shooting.

29. For most ACPR measurements, select *conservative*.

30. Select the fast envelope level you desire. This is available only when harmonic balance is chosen for the engine. When the fast envelope mode is set to off (default), normal envelope is performed at the transistor level. Harmonic Balance or Shooting is used to solve for a subset of the RF cycles. Level 1 is available for circuits where acceleration is desired, and the circuit has little memory effect Level1 requires defining the I and Q sources in the circuit, and the nodes to save data for in the circuit.

31. Select whether you want to have all the harmonics on the output along with the time-domain waveform or not. The default in fast envelope is to not calculate the time-domain output waveform and higher harmonics in order to save time. If you take the default, only the harmonic number you specify in the *Output harmonic* field is calculated. If you want all the harmonics and/or the time-domain waveform, select the *yes* check box to the right of *Output time-domain waveform and all harmonics*.

If you are using the port component with an I and Q modulation file, that port should be selected for both the I and Q source. If you are using the *ACPR_source*, descend-read

© 1994-2020 All Rights Reserved.

into the *ACPR_source* schematic. The I source is highlighted in red below, and the Q source is highlighted in blue below.

The mathematical model can be saved by setting the `writeenv` option to a file name. After this, the model file can be read in and used directly by setting the `readenv` option. You cannot set both options at the same time.

© 1994-2020
All Rights Reserved.

Now run the simulation. When the simulation completes, the psd (frequency response) plots in the waveform window.

The unit for the psd is dBW (dB with respect to 1 Watt).

© 1994-2020
All Rights Reserved.

The ACPR is shown under the Maestro tab of the ADE Explorer window.

Plotting EVM

EVM is integrated into the Direct Plot Form for the 802.11 standards only.

1. In ADE Explorer, select *Results - Direct Plot - Main Form*.
2. Select *envlp* results.
3. Select *EVM*.
4. Select the first harmonic.
5. Select *OFDM*.
6. If you are using rfVsource, select the standard you are using in rfVsource.

© 1994-2020 All Rights Reserved.

7. Set the symbol start to 0 (zero) except for 802.11a, which is 4u.
8. Select the output net in the circuit.

The EVM for both the data channels and the control channels is displayed.

Plotting Main Channel Power

Plotting main channel power is a bit less straightforward than plotting EVM.

1. In ADE Explorer, select *Results - Direct Plot - Main Form*.
2. Select *envlp* results.
3. Select *Main Channel Power*.
4. Select the first harmonic.
5. Type the bandwidth of the main channel.
6. Open the envelope *Choosing Analyses* form.
7. Open the ACPR wizard.

© 1994-2020 All Rights Reserved.

8. In the ACPR wizard, triple-click the entry in the *Symbol Start* field.
9. In the *Direct Plot Form*, click the middle mouse button in the *From* field. The entry from the ACPR wizard should transfer.
10. In a similar manner, transfer the Strobe Period, Window Size, and Repetitions entries.
11. Select the same window function that was used in the ACPR wizard.

© 1994-2020
All Rights Reserved.

12. Select the output net in the schematic. The power calculation is displayed with a marker in the display tool.

Plotting the Envelope Voltage Versus Time

1. In the ADE Explorer, select *Results - Direct Plot - Main Form*.
2. Select *envlp* results.
3. Select *Voltage*.
4. Select harmonic time.
5. Select the appropriate modifier.

© 1994-2020 All Rights Reserved.

6. Select the first harmonic.

© 1994-2020
All Rights Reserved.

7. Select the net in the schematic.

© 1994-2020
All Rights Reserved.

8. Zoom in if you would like. The symbols are turned on in the display below so the strobe points are visible.

Plotting the Time-Domain Waveform

This is available in fast envelope *level1* and *level2* only when *yes* is selected for Output time-domain waveform and all harmonics.

1. In ADE Explorer, select *Results - Direct Plot - Main Form*.
2. Select *envlp* results.
3. Select *Voltage*.

© 1994-2020 All Rights Reserved.

4. Select *time*.

© 1994-2020
All Rights Reserved.

5. Select the net in the schematic.

© 1994-2020
All Rights Reserved.

6. Zoom in, if desired.

The line that is visible between the spikes can be used to determine the phase of the signal as time increases only if the carrier frequency is an integer multiple of 1/delta-T in the I and Q modulation file. When the line is near the middle, the phase is zero or 180 degrees. When the line is near the top, the phase is 90 degrees. When the line is near the bottom, the phase is 270 degrees.

© 1994-2020 All Rights Reserved.

When you zoom in further, you can see the individual carrier cycles that envelope is calculating.

To see the full transient output, set the *transtrobeperiod* option to the period of the carrier divided by an integer. The choppiness of the waveform and the runtime is controlled by the

© 1994-2020 All Rights Reserved.

number of time points in the waveform that gets calculated. This is available in harmonic balance transistor-level simulation only. An example is shown below.

© 1994-2020
All Rights Reserved.

When zoomed in, the waveform is continuous.

Frequency Modulated Input Signals

Frequency modulated signals can be simulated in transistor-level envelope simulations only. When frequency modulation is used, the period of the waveform varies as a function of the modulation that is applied. Conventional envelope simulation cannot be used because the fundamental assumption is that the period is constant and defined by the carrier signal.

To allow frequency modulation, a new option has been added that allows the period to be solved at each carrier cycle that is simulated.

To simulate a frequency modulated signal, follow these steps:

© 1994-2020 All Rights Reserved.

1. Set up the frequency modulated source. In this example, a simple sinusoidal modulated port is used. If you use this element, remember that the modulation index is in percent.

© 1994-2020
All Rights Reserved.

2. Set up for a simulation using the envlp *Choosing Analyses* form and the ACPR wizard as shown before. Choose a standard with a bandwidth that is slightly larger than the FM modulation frequency. This will set the simulation to produce data at the time interval needed for the psd calculation. See the previous section for the steps that are needed.
3. Click *Options* in the envelope *Choosing Analyses* form.
4. Select the *Misc* tab.

5. Select the check box to the left of the number 1 for the *fmspeedup* option.
6. Type the modulation frequency in the *modulationbw* field.

© 1994-2020
All Rights Reserved.

7. Click *OK*.

8. In the envelope *Choosing Analyses* form, type in the center frequency in the *Fundamental Frequency* field.

9. Set harmonics and oversample as required. For more information, see the harmonic balance section at the beginning of Chapter 3.

10. Choose an accuracy level.

© 1994-2020 All Rights Reserved.

11. Set the *Fast envlp* mode to *off*.
12. Click *OK* and run the simulation.
13. When the simulation completes, the psd will plot.

© 1994-2020
All Rights Reserved.

14. If you want to see a spectral plot, set the line style to spectral. To do this, select the legend at the upper left. Right-click and select *Type* from the context menu. In the popup window, select *Spectral*. The line style will change to a more familiar plot.

15. When you zoom in to the individual spikes, you will see multiple spikes. There will be either one or two phantom spikes on each side of the output sideband because of the DFT sample window used in the psd function. The number of spikes and the relative

© 1994-2020 All Rights Reserved.

amplitude depend on the actual window function you selected. Remember that these spikes are not real. They are produced because of the window function itself.

Commonly Used Options

Options are provided to control the specific behavior of the simulation. This next section discusses the effect of the options in the options form for envelope. They are listed as they appear on the *Options* form.

© 1994-2020 All Rights Reserved.

Convergence Tab

ic (Shooting and Harmonic Balance)

Initial conditions can be specified graphically by selecting *Simulation - Convergence Aids - Initial Condition* in the ADE Explorer. Initial conditions can also be specified from a file using the *readic* property. Capacitors and inductors have the initial condition properties in the property list for the component. For capacitors, this is an initial voltage across the capacitor, and for inductors, it is an initial current in the inductor. The default is to observe all the initial conditions in the DC analysis that is used as the time-zero time point. The initial conditions force a voltage or current to be present in the time-zero solution. The initial conditions are released for the rest of the transient simulation in the tstab interval. The *ic* option controls the initial conditions that should be observed in the time-zero time point. *all* is the default. *dev* means that only the initial conditions on capacitors and inductors are observed. *node* means that only the initial conditions on a node are observed. *dc* means that no initial conditions are observed. The example below shows *node*.

© 1994-2020
All Rights Reserved.

Skipdc (Shooting and Harmonic Balance)

In some cases, the time-zero time point DC analysis does not converge. Instead of stopping the simulation, *skipdc* allows the envelope simulation to continue using an assumed solution for the time-zero time point. The default is *no* and a DC analysis is run to get the initial time point. *yes* means skip the DC solution, and proceed directly to the envelope simulation. All the nodes with initial conditions specified start at the initial condition value. Nodes with batteries start at the battery voltage. Nodes with no initial conditions start at zero volts. For `skipdc=yes`, the signal sources start as specified immediately in the envelope simulation. The example below shows the *skipdc* option set to *no*. The default is *yes*.

Readic (Shooting and Harmonic Balance)

This specifies an ascii file that contains two columns to be read as initial conditions. The left column is the node name. The right column is the voltage value. If the entry does not start with / (slash), the entry is located in the netlist directory. To find the netlist directory, select *Setup- Simulator/Directory/Host* in ADE Explorer. Look in the *Project Directory* field for the location of the simulation directory. Navigate to that directory and then to the `<Circuit Name>/spectre/<schematic or config>/netlist` directory. You can also click (...) and browse to the directory. There is no default.

Readns (Shooting and Harmonic Balance)

This specifies an ascii file with the same format as an ic file that is used as nodesets for the time-zero DC solution. Nodesets do not force a voltage to be held for the time-zero solution. Instead, they are a way of speeding up the time-zero calculation. As a suggestion, set the *write* option and the *readns* option to the same filename. The *write* option writes the time-zero solution to a file. When this is used as a starting point, many fewer iterations are needed for the time-zero point to converge. There is no default.

Cmin (Shooting and Harmonic Balance)

cmin adds a capacitor to ground from all nodes in the circuit to ground with the value specified in the *cmin* field. Note that if 1 is specified, a 1 farad capacitor is added from every node to ground in the circuit. Do not forget the multiplier suffix. Values from 1f to 100f are typically used to allow convergence. The default is 0.

Maxiters (Shooting and Harmonic Balance)

The transient analysis for tstab (for both shooting and hb) and shooting interval iterate to a solution at each time point. *maxiters* specifies the maximum number of iterations before the

© 1994-2020 All Rights Reserved.

timestep is cut for another try at convergence. In some cases, model parameters can cause discontinuities in the device current or capacitance. If this occurs, the change in the circuit condition can be large enough to require more than the five iterations that are the default. Specifying *maxiters* in the 40 to 100 range usually allows the simulator to converge in spite of the discontinuity.

restart (Shooting and Harmonic Balance)

This option is used when a sweep is run. If restart is *no*, the time-zero time point does not restart from scratch. It restarts from the time-zero solution of the previous sweep value. *yes* forces a new time-zero solution to be re-run from scratch every time. The default is *no*.

envmaxiters (Shooting and Harmonic Balance)

The solution for each period of the carrier is an iterative solution. *envmaxiters* sets the maximum number of iterations that are allowed before the number of periods skipped is reduced. The default is three iterations for shooting and 40 for harmonic balance.

Accuracy Tab

© 1994-2020 All Rights Reserved.

method (tstab and shooting window for shooting, tstab only for Harmonic Balance)

This option controls the integration method for the shooting window. Note that in harmonic balance, the *method* option only applies in the tstab interval because the harmonic balance envelope solver works in the frequency domain. The default value for the shooting window is determined by the setting of *errpreset*. For *moderate* and *conservative*, the default is *gear2only*. For *liberal* it is *traponly*. Generally, *gear2only* is preferred because of the absence of trapezoidal ringing inherent in the *trap* (trapezoidal) method. Note that *liberal* is not recommended for RF simulation. When you simulate high Q oscillators, set *method* to *traponly*. This is needed because *gear2only* slightly numerically damps the oscillations.

envmethod (Shooting and Harmonic Balance)

This option controls the integration method for the part of the envelope analysis during the skipped cycles. In this part of the simulation, numerical integration still needs to be performed to get correct answers. For example, if there is a long time constant, the behavior on the node with the long time constant needs to be integrated even during the skipped cycles. *envmethod* controls this. The default is *gear2only* except for autonomous envelope (oscillators) where the numerical damping of *gear2only* is unacceptable. In this case, the default is *traponly*, which neither damps nor exaggerates the oscillations.

relref (Shooting (tstab and shooting window) and Harmonic Balance (tstab only))

In the order of most to least accurate, the *relref* settings are *pointlocal*, *alllocal*, *sigglobal*, *allglobal*. The default for *moderate* is *sigglobal* and for *conservative* it is *alllocal. relref* is used in the transient analysis in the tstab interval for both shooting and harmonic balance and in the shooting window for shooting. *relref* is used in hb only for the tstab interval because the solver works in the frequency domain. In some cases, the timestep can collapse to near zero. The symptom is that in the Spectre output window, the percent done number remains the same for many reporting intervals. The solution to this is to set *relref* to *sigglobal*, which is slightly less accurate. Another possibility is to use the *lteminstep* option as described in the *AdditionalParams* option of the *Misc* tab.

lteratio ((tstab and shooting window for shooting) and Harmonic Balance (tstab only))

lteratio is a multiplier for the allowable numerical integration error in the tstab interval and the shooting window. The default is 3.5 or 10 depending on the setting of *reltol* and *errpreset*. See Controlling Accuracy on page 672 for details. *lteratio* cannot be set smaller than 1.0 and is normally between 3.5 and 100. In some cases, the timestep collapses to near zero and setting *relref* to *sigglobal* does not fix the problem. In this case, there might be a model discontinuity. To test this, disable numerical integration timestep control by setting *lteratio* to 1e9. Note that this is an extreme measure not to be used under normal simulations. If the

© 1994-2020 All Rights Reserved.

timestep still collapses, there is a discontinuity in one of the models in the circuit. This could be a device model or a Verilog-A model. If it is necessary to set *Iteratio* very large to get the simulation to complete, you must also set *maxstep* small enough to preserve the accuracy of the simulation. Note that *Iteminstep* might also be used to treat this condition. See *AdditionalParams* in the *Misc* tab to see how to use this option.

envIteratio (Shooting and Harmonic Balance)

envIteratio is a multiplier for the allowable numerical integration error in the numerical integration that occurs in the part of the waveform where cycles are skipped. The default is 0.35 for conservative and 3.5 for moderate. Liberal is not recommended.

steadyratio (shooting Only)

The maximum mismatch between the beginning and ending values for the voltages and currents in the shooting window is determined by the *convergence criteria * Iteratio * steadyratio*. The default is 0.1 for moderate, and 1.0 for conservative.

sweepmethod

When fast envelope is used, a series of harmonic balance simulations are run. The amplitude is swept over a range of values present in the I and Q modulation files. When the baseband model is used in fast envelope, the phase also needs to be swept. The sweepmethod option controls the density of the sweep points. The default is coarse, where 7 amplitude and 12 phase points are used. This is sufficient for most simulations. Fine uses 10 amplitude points and 16 phase points. Userdefined allows you to specify the number for both magnitude and phase points. Adaptive is a work in progress.

Magnitude points

When sweepmethod is set to userdefined, Magnitude points specifies the number of different amplitude points to use in the characterization phase of fast envelope.

Phase points

When sweepmethod is set to userdefined, and there is a baseband source in the circuit, this option controls the number of phase points to use in the characterization phase of fast envelope.

© 1994-2020 All Rights Reserved.

Output Tab

annotate

Annotate (Shooting and Harmonic Balance)

This option controls the level of detail in the output log. The default is *status.* The level of detail increases as you move to the right.

Save (Shooting and Harmonic Balance)

The default is *allpub*. This saves all the public voltages at all levels of the hierarchy in the schematic. It excludes the internal nodes of the device models. *all* adds the internal nodes of all the devices. *lvl* saves all the nodes including the internal nodes of the devices through the level of hierarchy set in the *nestlvl* option. *lvlpub* is like *lvl*, but it does not save the internal nodes of the devices. *selected* saves only the nodes that are specifically saved. In ADE Explorer, this is accomplished using the *Outputs - To Be Saved - Select On Schematic* menu option and then selecting the nodes and terminals specifically in the schematic. At the netlist level, this is accomplished by using a save statement with a list of names to be saved.

© 1994-2020 All Rights Reserved.

Nestlvl (Shooting and Harmonic Balance)

If *save* is set to *lvl* or *lvlpub*, this controls the maximum level of hierarchy to be saved. If *nestlvl* is *1*, only the top level is saved. If *nestlvl* is 2, the top level and the next level down are saved. The value for *nestlvl* can be any integer. There is no default for *nestlvl*.

Compression (Shooting and Harmonic Balance)

Do not use this option.

outputtype (Shooting and Harmonic Balance)

This option is rarely used. It controls which simulation results to save. Normally, both time and frequency domain data is saved. Envelope saves the time-domain data only. Spectrum saves just the frequency-domain data. The default is to save both time and frequency-domain data. Note that in fast envelope, you need to select the *yes* check box to the right of *Output time-domain waveform and all harmonics* in order to calculate time-domain output.

strobeperiod (Shooting and Harmonic Balance)

This option specifies the envelope time in seconds to force cycles of the carrier to be simulated. Shooting will round this time to an integer number of periods of the carrier. This is used to force data at the times needed for the psd calculation. There is no default value.

transtrobeperiod (Shooting and Harmonic Balance)

Normally, the time-domain waveform is a cycle of carrier followed by many skipped cycles, followed by another cycle of simulation. Since there is no waveform between the last point for the first cycle and the first point of the second cycle, the waveform tool just connects those points together with a straight line. The waveform can be made into a continuous waveform using the *transtrobeperiod* option. This should be set to an integer fraction of one period of the carrier frequency. When this is set, Spectre still skips cycles, but it also creates an interpolated, continuous waveform with time points at the *transtrobeperiod* interval. In the example below, 20 points are set for one period of the carrier at 2.5G.

Spectre does take expressions, as shown above. There is no default value for *transtrobeperiod*.

© 1994-2020 All Rights Reserved.

Here is a comparison of the 2 waveforms.

© 1994-2020
All Rights Reserved.

The red waveform is the normal waveform. The vertical spikes in the red waveform are the individual periods of the carrier that are normally calculated. When *transtrobeperiod* is set, the waveform becomes continuous, as shown in blue. A closer view is shown below.

© 1994-2020
All Rights Reserved.

Reuse Tab

Write (Shooting and Harmonic Balance)

write specifies a filename to which the DC solution used as the first time point be written to. If the name does not start with slash (/), the file is written in the *netlist* directory. To determine the netlist directory location, select *Setup - Simulator/Directory/Host* in ADE Explorer. The *Project Directory* field lists the location of the simulation directory. From this directory, the netlist directory is in `<Circuit Name>`/`<simulator name>`/*<schematic or config>/netlist*. There is no default value.

Writefinal (Shooting Only)

writefinal writes the final envelope time point from the last period of the envelope analysis (not the final time point of the tstab interval) to the filename specified. See the description for *write* for the location. There is no default value.

Swapfile (Shooting Only)

This option is not currently implemented.

© 1994-2020 All Rights Reserved.

Saveclock (Shooting and Harmonic Balance)

This option is not currently implemented.

Saveperiod (Shooting and Harmonic Balance)

This option is not currently implemented.

Savetime (Shooting and Harmonic Balance)

This option is not currently implemented.

Savefile (Shooting and Harmonic Balance)

This option is not currently implemented.

Recover (Shooting and Harmonic Balance)

This option is not currently implemented.

Writeenv

When fast envelope is used, the behavioral model can be written to a file and reused. Writeenv specifies the filename to save the behavioral model for fast envelope in.

Readenv

When fast envelope is used, the behavioral model can be written to a file and reused. Readenv specifies the file to read back in instead of running the characterization from scratch.

© 1994-2020 All Rights Reserved.

Misc Tab

start (Shooting and Harmonic Balance)

This option is rarely used. It defines the starting time for the envelope simulation. The default is 0.

© 1994-2020
All Rights Reserved.

outputstart (Shooting and Harmonic Balance)

This option is rarely used. This defines the simulation time to begin to save data to the output file. The default is to start writing data at time 0.

tstab (Shooting and Harmonic Balance)

This is seldom used in envelope simulation. This specifies a simulation time for the transient analysis to run before starting the envelope simulation. The default is to run 3 periods of the carrier signal.

modulationbw (Shooting and Harmonic Balance)

modulationbw and *envmaxstep* accomplish the same thing, which is to define a maximum time interval to skip between carrier cycles that are simulated. When *modulationbw* is set, at least eight periods of the carrier are simulated in the time period of one cycle of the frequency *modulationbw*. *modulationbw* does not force data at equally spaced intervals, like *envstrobeperiod* does. Instead, it sets a a maximum delta-time for skipping cycles. There is no default.

maxstep (Shooting: tstab and waveform calculation. Harmonic balance: tstab interval only)

This option is rarely used. This sets the maximum delta-time for the shooting waveform in each cycle of the carrier. The default forces at least 20 time points for moderate, and 50 time points for conservative. It also is used in the transient analysis in the tstab interval.

envmaxstep (Shooting and Harmonic Balance)

envmaxstep and *modulationbw* accomplish the same thing, which is to define a maximum time interval to skip between carrier cycles that are simulated. *envmaxstep* sets the maximum time between periods of the carrier that can be skipped. *envmaxstep* does not force datapoints at regularly spaced intervals. Instead, it sets a a maximum delta-time for skipping cycles. The default forces at least 25 periods to be simulated when *moderate* is selected, and 50 when *conservative* is selected.

pactimes (Shooting and Harmonic Balance)

During the transistor-level envelope simulation, one or more of the carrier periods that are simulated can be used as the nonlinear operating point for the small-signal pac, pxf, pnoise,

© 1994-2020 All Rights Reserved.

or psp analyses. *pactimes* is a list of times separated by a space for the large-signal envelope solution that should be used for the small-signal analyses. The default is to not use any points for the small-signal analyses.

pacnames (Shooting and Harmonic Balance)

This option is used in combination with *pactimes*. Specify a list of the small-signal analyses pac, pxf, pnoise, and/or psp in this option field, and these analyses will be run at the times specified in *pactimes*. The default is to not run any small-signal analyses.

When this option has a list of small-signal analyses, the corresponding pac, pxf, pnoise, and psp analyses must also be set up and enabled.

fmspeedup (Shooting and Harmonic Balance)

This option allows the period of the carrier cycle to vary because of frequency modulation. When *fmspeedup* is set to the default of 0, the period of the carrier is not allowed to vary. When it is set to 1, the period is allowed to vary.

AdditionalParams (Shooting and Harmonic Balance)

This is a field for `<option_name>=value` statements. Multiple `<option_name>=value` statements are allowed with a space between them. Generally, this is used to unlock the beta features, however, there is one case where this field might be useful for an option that is not available in the GUI.

In some cases, in the tstab interval, the timestep collapses to near zero. The symptom is that the percent complete stays at the same value for many reporting intervals. In this case, there might be a discontinuity in one of the models.

In transient simulation, two things can reduce the timestep; having too much numerical integration error or taking too many iterations at a single timestep can cause the timestep to be cut. Spectre does not report what is causing the timestep to become small. One way to eliminate the possibility of numerical integration error (which is called local truncation error in the simulator) as the cause of reducing the timestep is to set *Iteratio* to 1e9. The disadvantage of this is that it disables the normal method of timestep control, therefore, *maxstep* must be set to maintain accuracy.

© 1994-2020 All Rights Reserved.

Another way to accomplish this is to set *lteminstep* to a value of about the stop time divided by 1e5. This allows the normal method of timestep control, but if for some reason the numerical integration error wants to cut the timestep too small, the error is ignored. *lteminstep* is the way to specify this. To do this in ADE Explorer, type `lteminstep=<Stop_Time>/1e5`. If you use `lteratio=1e9`, or if you set *lteminstep* and you still have very small timesteps, try increasing *maxiters* in the *Convergence* tab to between 40 and 100. When *lteratio* or *lteminstep* is set and you still have timestep problems, there is likely a discontinuity in either a device model or in a Verilog-A model. An example is shown below.

Autonomous ENVLP Analysis (Oscillators)

Autonomous envelope is useful for simulating the startup behavior of high Q oscillators, and for simulating the frequency pulling and/or pushing effects caused by random data being applied to a system. In autonomous envelope, the period and amplitude of each cycle of the oscillator that is simulated needs to be solved.

In general, harmonic balance is suggested for LC oscillators, and shooting should be used for ring oscillators. Crystal oscillators should use harmonic balance.

The only additional requirement in the *Choosing Analyses* form for oscillators is the requirement to select the check box for oscillators, and specify a node or pair of nodes that has the oscillator signal on it. If the *osc initial condition* check box is set to *linear*, the nodes must be in the resonator of the oscillator. Although not a requirement, it is recommended that you use adaptive timestep control, and set the *modulationbw* (modulation bandwidth) option to the frequency of the oscillator divided by the Q of the oscillator.

© 1994-2020 All Rights Reserved.

Example: Startup behavior of a crystal oscillator

Here is an example circuit.

© 1994-2020
All Rights Reserved.

First, add an isource, set to *Source type* to *pulse*, and connect it to the resonator as highlighted above. The properties list is shown below.

Next, set the properties.

© 1994-2020
All Rights Reserved.

1. Set the *Zero value* to 0 (zero). The pulse starts at the zero value. With zero amps, the isource is a perfect open, so it does not load the circuit.
2. Set the *One value* to a current that starts the oscillator, but does not drive it too hard at the start. You will need to experiment with this value using the transient. In the waveform window below, the junction of the inductor and capacitor in the crystal motional equivalent circuit is plotted. This voltage will get quite large because of the high Q. Change the *One value* so that one to ten volts peak is generated at this node at the start of the transient simulation, and no huge transients are generated.

3. Set the *Rise time* and *Fall time* to about 1/10th to 1/100th of the period. This is a 10MHz crystal oscillator. In this case, 1/100th of the period was used.
4. Set the *Pulse width* so that half a period is generated by the pulse. Half the period of 10M is 50nsec, so the total of *Rise time*, *Fall time*, and *Pulse width* is 50nsec.
5. Do not make an entry in the *Period of waveform* property.

The effect of these entries is to set up a single current pulse to start the oscillator.

© 1994-2020 All Rights Reserved.

Now, set up the envelope *Choosing Analyses* form.

1. Select the appropriate engine. For LC and crystal oscillators, select *Harmonic Balance*. For ring oscillators select *Shooting*.

© 1994-2020
All Rights Reserved.

2. Set an appropriate *Stop Time* for your simulation. The time chosen will depend on the Q of the oscillator. About five times the Q periods need to be simulated.
3. Set the *Fundamental frequency*, *Number of Harmonics*, and *Oversample Factor* as needed for your simulation. See the discussion about setting harmonics and oversample factor in *Chapter 3: Harmonic Balance*.
4. Select *adaptive Time Step Control*. This allows the simulator to place the periods of oscillator simulation at the appropriate place.
5. Enable the check box next to the *Oscillator* Property, and specify a node inside the resonator for the oscillator node(s).
6. Set *Osc initial condition* to *default*. This ensures that the oscillator starts up as a result of the current pulse applied to the resonator.
7. Select *moderate* accuracy. In virtually every case, this is accurate enough.
8. *Fast envlp mode* must be set to *off*. Fast envelope is not available for oscillators.

Options are not required, but are suggested. Click *Options* at the bottom of the *Choosing Analyses* form.

Set the *modulationbw* (modulation bandwidth) property to the period divided by the Q of the oscillator. Expressions are allowed by Spectre.

© 1994-2020 All Rights Reserved.

Now run the simulation. Below is a comparison of the runtimes for the envelope analysis and for the transient analysis. Note that the runtime is much shorter for the envelope analysis.

```
/servers/scratch03/kamal/spectrerf131/testcase131/docs_database
File  Edit  View  Help                                          cādence
EnvStepIndex = 553, envTime = 9.85292 ms, completed = (98.5 %)
Current Frequency = 9991359.78305946663
EnvStepIndex = 554, envTime = 9.87265 ms, completed = (98.7 %)
Current Frequency = 9984289.27985462174
EnvStepIndex = 555, envTime = 9.89238 ms, completed = (98.9 %)
Current Frequency = 9981939.36349599808
EnvStepIndex = 556, envTime = 9.9023 ms, completed = (99 %)
Current Frequency = 9982476.35819943808
EnvStepIndex = 557, envTime = 9.92203 ms, completed = (99.2 %)
Current Frequency = 9989433.86748934351
EnvStepIndex = 558, envTime = 9.93184 ms, completed = (99.3 %)
Current Frequency = 9986377.97248011827
EnvStepIndex = 559, envTime = 9.95156 ms, completed = (99.5 %)
Current Frequency = 9996838.53402140364
EnvStepIndex = 560, envTime = 9.97136 ms, completed = (99.7 %)
Current Frequency = 10001410.4670494106
EnvStepIndex = 561, envTime = 9.99115 ms, completed = (99.9 %)
Current Frequency = 10008519.5101154502
Frequency = 9.996839e+06 (Hz)

Total clock cycles: 99928, skipped cycles: 99366, speed-up factor: 154
Done with the envelope-following analysis.
Clean up envelope following analysis.
Total time required for envlp analysis `envlp': CPU = 33.8629 s, elapsed = 34.3503 s.
Time accumulated: CPU = 34.2848 s, elapsed = 35.221 s.
Peak resident memory used = 46.3 Mbytes.


Notice from spectre.
    260 notices suppressed.
    310 warnings suppressed.
5  >                                                    L1,437   C62
```


© 1994-2020 All Rights Reserved.

© 1994-2020 All Rights Reserved.

Now use the *Direct Plot Form* to plot the waveform. In ADE Explorer, select *Results - Direct Pot - Main Form*.

1. Select *envlp* results.
2. Select *Voltage*.
3. Select *Net* or *Differential Nets*.
4. Select *time*.

© 1994-2020 All Rights Reserved.

5. Select a net in the schematic. The waveform is displayed.

The waveforms above are the envelope results on the left, and the transient results on the right. The results are almost identical for both, and envelope takes 16.4 seconds versus 2125.4 seconds for the transient on the same machine.

Common Problems With Oscillator Startup

In some cases, the oscillator will have a hard time starting up. You can recognize this by looking at the waveform in the resonator of the oscillator. Normally, the junction of the capacitor and the inductor in the crystal equivalent circuit will have voltages in the order of

© 1994-2020 All Rights Reserved.

KiloVolts peak at steady-state. When you see the ragged startup waveforms, as shown below, set a fixed timestep with about 15 to 100 periods as the step interval.

© 1994-2020
All Rights Reserved.

Spectre takes expressions in the entries, so it makes the period calculation easy.

The example above sets 20 periods of 10MHz as the step interval.

© 1994-2020
All Rights Reserved.

Also, set the following analysis options:

Set the *method* and *envmethod* options to *traponly*. *trap* does not damp or emphasize the oscillations of the oscillator. The default is *gear2only*, which slightly damps the oscillations. Also, set *relref* to *alllocal*. This is needed because of the mix of very large amplitude signals on the resonator with much smaller oscillations in the rest of the circuit.

Also, in some circuits, the initial pulse that is used to start the oscillator can cause large transients at the beginning of the envelope analysis and cause problems with the

© 1994-2020 All Rights Reserved.

determination of the frequency at the beginning of the simulation. To determine if this is the case, look at the startup waveform at the beginning of the simulation.

© 1994-2020
All Rights Reserved.

Note that the waveform is still settling at the beginning of the simulation. In this case, four periods are being simulated instead of one. You can see the frequency that is being used in the output log.

© 1994-2020 All Rights Reserved.

Setting *tstab* in the *Misc* tab can cause the simulation to start robustly.

© 1994-2020
All Rights Reserved.

The waveform is much smoother with tstab set.

© 1994-2020
All Rights Reserved.

Examples

There are wireless examples provided with the release. To access the examples:

1. Navigate to `<mmsim_install_directory>/tools.<platform>/spectre/examples/SpectreRF_workshop`.
2. Copy the `rfworkshop.tar.Z` file to a directory where you have write access, and extract the file.
3. Navigate to the directory `/spectrerf_workshop/doc`.

In this directory, you will find directions for the wireless workshop.

© 1994-2020 All Rights Reserved.

© 1994-2020 All Rights Reserved.

7

Large-Signal S-Parameter Simulation (LSSP)

Large-Signal S Parameters

Characterizing RF circuits with small-signal S-parameters is a well established practice. However, small-signal S-parameters are not sufficient to describe nonlinear circuits with moderate to large amplitude signals.

As a natural extension of small-signal S-parameters, large-signal S-parameters (LSSPs) are defined as the ratio of reflected (or transmitted) waves to incident waves. Since small-signal S-parameters are based on the simulation of a linearized circuit, they are independent of the input power. LSSPs include the nonlinear effects at different amplitudes.

The LSSP capability is implemented for single-tone harmonic balance and PSS Shooting or harmonic balance simulations.

Starting with the MMSIM13.1 release, the LSSP capability has been integrated into the hb and pss (shooting and hb) *Choosing Analyses* form. With this integration, the simulation is now much easier to set up, and allows the input and output frequencies to be different. Direct plot functions have also been implemented to allow plotting the large-signal S-Parameters on a Smith Chart. When these expressions are added under the Maestro tab of the ADE Explorer window from the *Direct Plot Form*, ADE Explorer plots the output on the Smith Chart.

© 1994-2020 All Rights Reserved.

Large-Signal S parameters for a Two Port Circuit

Consider calculating LSSPs for a two-port circuit. When you apply a signal at port 1 on the left and terminate port 2 on the right, as shown below, you can measure S_{11} and S_{21}.

For a wave A launched from port1 on the left, S_{11} is B/A, and S_{21} is C/A, where A, B, and C are all vectors. When this measurement is done, the power in the load resistor is calculated.

© 1994-2020
All Rights Reserved.

Similarly, when you launch wave D at port 2 on the right, and terminate port 1 on the left as shown below, you can measure S_{12} and S_{22}. The power measured above is set in port2 for this measurement.

For a wave launched from port2 on the right, S_{22} is E/D, and S_{12} is F/D, where D, E, and F are all vectors.

For nonlinear circuits, the values of the S-Parameters can change as the amplitude of the signal changes. In addition, the frequency can be swept for a single input and output amplitude at a time

Circuit Setup

The normal setup for large-signal S-Parameter measurement is similar to the setup for a swept input power simulation, with the addition of setting the output *Source type* to *sine* and leaving all the properties for the sine wave blank. LSSP analysis causes port 1 to be driven first so the S_{11} and S_{21} measurements can be made, and then causes port 2 to be driven next so the S_{12} and S_{22} measurements can be made. The power in port 2 from the S_{21} measurement is set in port 2 for the S_{22} and S_{12} measurement.

© 1994-2020 All Rights Reserved.

A behavioral low noise amplifier is shown below.

LSSP expects that a variable is used to set the input power. The normal setup is to specify the input power sweep as normal and then select the LSSP check box in the *Choosing Analyses* form. When this is chosen, you define the source and load ports in the *Ports* field, and you select the frequency of the output in the *Load Harmonic* field. The output frequency defines the frequency for the forward S_{21} measurement, and the frequency of port 2 for the S_{22} and S_{12} measurements. Set the output *Source type* to *sine* with no properties to define the frequency or amplitude.

© 1994-2020
All Rights Reserved.

The input port RF and output port *PORT2* setup are shown below.

© 1994-2020
All Rights Reserved.

Note that the input source is set to produce two RF signals. Both signals have frequencies that are set by the variables *frf1* and *frf2*. The amplitude of both signals are set to the variable name *prf*. This is a common way to set sources because it allows changing the frequencies and amplitudes using variable names in ADE Explorer without needing to change the schematic.

The normal setting for the load port is to set the *Source type* to *sine*, and leave all the properties that define the sinusoid blank, as shown above.

© 1994-2020
All Rights Reserved.

ADE Explorer Setup

The ADE Explorer setup is shown below with the variables defined in the *Design Variables* section.

Note that the variable *frf2* has been set to zero to disable the second RF tone.

Analysis Setup

Next, set up the hb *Choosing Analyses* form, as shown below.

1. Select *Analyses* - *Choose* from the ADE Explorer window.

© 1994-2020 All Rights Reserved.

The *Choosing Analyses* form is displayed, as shown below.

© 1994-2020
All Rights Reserved.

2. Specify the fundamental frequency in the *Fundamental Frequency* field. Here, 900M has been specified.
3. Specify the number of harmonics in the *Number of Harmonics* field. Here, 15 is specified.

   It is recommended to set the number of harmonics manually and specify a value that is high enough to get accurate results at the highest input power level in the sweep.
4. Select the LSSP check box. The LSSP options are displayed, as shown below.

5. Click *Select* to the right of the *Ports* field and select the input source and output source from the schematic. The *Ports* field is populated with the names of the input and output ports, as shown below.

6. Click *Choose* to the right of the *Load Harmonic* field and select the frequency and tone, as shown below.

7. Click *Apply*.

© 1994-2020
All Rights Reserved.

LSSP Amplitude Sweep

To set up the amplitude sweep:

1. Select the *Sweep* check box. The Sweep options are displayed, as shown below.

© 1994-2020
All Rights Reserved.

2. Click *Select Design Variable*. The Select Design Variable form is displayed, as shown below.

3. Select a variable from the available list of variables. The *Variable Name* field is populated with the name of the variable, as shown below.

© 1994-2020
All Rights Reserved.

4. Specify the sweep range in the *Start* and *Stop* fields.
5. Specify the step size in the *Step Size* field.
6. Click *OK*.
7. Run the simulation.

Plotting the S11 Curve

To plot the S_{11}curve:

© 1994-2020
All Rights Reserved.

1. In the ADE Explorer window, choose *Results - Direct Plot - Main Form*. The *Direct Plot Form* is displayed, as shown below.

2. Choose *Z-Smith* from the *Plot Type* section.

3. Select the *Add to Outputs* check box to add the expression in the *Outputs* section of the ADE Explorer window.

4. Click the *S11* button.

   The input reflection coefficient is displayed on the Smith chart, as shown below.

© 1994-2020 All Rights Reserved.

© 1994-2020
All Rights Reserved.

5. Double-click the trace legend. The *Complex Trace Properties for LSS11* window is displayed, as shown below.

6. Select the *Turn On Symbols* check box.
7. Select the *Show All points* check box.
8. Click *OK*.
9. In the waveform window, select *Marker - Snap Track Cursor* and move the mouse pointer over the trace legend.

© 1994-2020 All Rights Reserved.

The tracking cursor displays the input power level, as shown below.

Similarly, you can plot the S_{12} curve.

Plotting the Input Impedance Magnitude and Phase Curves

To plot the input impedance magnitude and phase curves:

1. In the *Direct Plot Form*, select *ZM* from the *Function* section.
2. Select *Magnitude* from the *Modifier* section.
3. Click *ZM1*.
4. Select *Phase* from the *Modifier* section.
5. Click *ZM1* again.

© 1994-2020 All Rights Reserved.

The impedance and phase curves are displayed in the waveform tool, as shown below.

© 1994-2020
All Rights Reserved.

© 1994-2020 All Rights Reserved.

8

AnalogLib Components Used in RF Simulation

The Delayline Element

The delayline element is a lossless transmission line with a specified delay time T_d and characteristic impedance Z_0. It is intended for the small-signal ac, sp, and noise analyses, and the harmonic balance analysis and models ideal delays in these analyses.

The ABCD matrix of a lossless transmission line section is given by:

$$ABCD = \begin{bmatrix} \cos(\omega T_d) & jZ_d \sin(\omega T_d) \\ \dfrac{j\sin(\omega T_d)}{Z_0} & \cos(\omega T_d) \end{bmatrix}$$

The delayline symbol is shown below.

Command-line help

spectre -h mtline

© 1994-2020
All Rights Reserved.

CDF Parameters

CDF Parameter Label	CDF Parameter	Spectre	auCDL	auLVS	hspiceD	Default
Characteristic Impedance	`z0`	X	-	-	X	50
Delay Time	`td`	X	-	-	X	1n

The properties list is shown below.

© 1994-2020
All Rights Reserved.

The PORT Element

You can use the *port* component, located in the *analogLib* library, in RF circuits for Spectre® circuit simulator RF Analysis (SpectreRF) and Spectre S-parameter simulations.

The *port* component, located in the *analogLib* library, is a superset of the existing *psin* component. The *port* component supports all the source types of the Spectre *port* primitive: *pwl*, *pulse*, *sine*, *dc*, and *exp.* In addition, it includes the *bit* and *prbs* source types. If you are using the `psin` component, it is highly recommended that you switch to the port component because the properties are organized in a better way, and the levels in volts peak and in dBm both cannot be set at the same time. This is easily accomplished by obtaining the properties form for `psin`, and changing the cell name to `port`. All the properties that are set in the `psin` component will retain their values when you make the change.

Note: The existing `psin` is retained for legacy designs and will not be enhanced in any way moving forward.

The *port* component is an independent resistive source tied between positive and negative terminals. It is equivalent to a voltage source in series with a resistor, where the reference resistance of the *port* is the value of the resistor.

Capabilities of the port Component

While the *port* component is most useful as a stimulus in high-frequency circuits, it also has the following unique capabilities.

- It can define the ports of a circuit to the S-parameter analysis.
- It has an intrinsic noise source that lets the noise analysis directly compute the noise figure of the circuit.
- Is the only source for which you can specify the amplitude in terms of power.
- It generates sinusoidal, exponential, piecewise-linear, periodic piecewise-linear, pulse, bit, and prbs waveforms for all the large-signal analyses.

© 1994-2020 All Rights Reserved.

- It can have different impedances for each harmonic number in the harmonic balance large-signal analysis.

Terminating the Port

When you specify the voltage on a *port*, all Spectre simulations assume that the port is properly terminated in it's reference resistance. The specified voltage value is not the voltage on the internal voltage source, which is actually set to twice the value specified on the `port`. If you use a `port` source to drive an open circuit, the voltage (for DC, transient, AC, and PAC signals) is double its specified value. However, you can alternatively specify the amplitude of the sine wave in the transient and PAC analyses as the power in dBm delivered by the *port* when terminated with the reference resistance. For the `bit` and `prbs` source, the voltages specified for the zero and one states are doubled inside the port as usual. For the `prbs` source, you can add a matching resistor across the port to get the voltages specified in the zero and one states, or you can omit the resistor, and half the desired voltages. For the bit source, if you intend to use the z state, do not add a matching resistor across the source because this small resistor will totally swamp the Z-state effect. Thus, when you use the bit source in a port, the values for the zero and one states should be half the desired voltages. This is not an issue for the `vsource` element.

The *port* component *Edit Object Properties* form is shown in Figure 8-1 on page 1717 and Figure 8-2 on page 1718.

© 1994-2020 All Rights Reserved.

Figure 8-1 Top of the port Component Edit Object Properties Form

© 1994-2020
All Rights Reserved.

Figure 8-2 Bottom of the port Component Edit Object Properties Form

Parameters for the Port Component

Port Parameters

Port parameters include *Resistance*, *Reactance*, *Port Number*, and *Multiplier*.

Resistance

The reference resistance of the system. The value must be a real number, but not 0. The default value is 50 ohms. This applies for all analyses. Note that the resistance and reactance can be set individually for each harmonic in harmonic balance large-signal analyses only. For more information, refer to Using the Harmonic Port with Harmonic Balance on page 1719.

© 1994-2020 All Rights Reserved.

Reactance

The imaginary part of impedance is used for harmonic balance large and small-signal analyses, and for the ac, noise, xf, sens, stb, pz, and sp.analyses. The reactance is not used for the dc, transient, and shooting-based large and small-signal analyses. The value must be a real number. The default is zero ohms. Note that the resistance and reactance can be set individually for each harmonic in harmonic balance large-signal analyses only. For more information, refer to Using the Harmonic Port with Harmonic Balance on page 1719.

Port number

The number associated with the *port*. The value must be a non-zero integer and must be unique for each *port* in a schematic. The port number is not automatically indexed when you place a new *port* on your schematic. There is no need to assign this number because in the sp analysis, you just select the port instances you want from the schematic.

If you do assign port numbers, start with the number one, and number the ports sequentially. A gap in the port number sequence causes a simulation error if the port list is not defined in the S-parameter analyses.

Multiplier

Leave this property at the default. The multiplicity factor specifies the number of *ports* in parallel. The value must be a non-zero real number and the default is 1. For example, if you set *Resistance* to 50 and *Multiplier* to 2, you specify two *ports* in parallel, with an effective reference resistance of 25 ohms.

Using the Harmonic Port with Harmonic Balance

The port component has a feature called harmonic port that allows the specification of different impedances at different harmonic numbers. This is available for harmonic balance large-signal analyses only and does not work with any other analysis.

© 1994-2020 All Rights Reserved.

For this example, a 2.4GHz LNA is the example circuit.

© 1994-2020
All Rights Reserved.

The input and output ports are set up normally, as shown in the properties forms below.

© 1994-2020
All Rights Reserved.

PORT1 is the input and *PORT2* is the output.

© 1994-2020
All Rights Reserved.

One Input Frequency

The harmonic balance *Choosing Analyses* form has a single input applied.

© 1994-2020
All Rights Reserved.

To use the harmonic port, select *Harmonic Port* on the properties form. In this case, the port being used as a load in the circuit is selected. The *Edit harmonic port* window is displayed.

For this setup, a single input frequency is applied to the circuit. In this case, harmonic numbers 1, 2, 3, 4, and 5 exist in the simulation result. In the *Harmonic* column, you enter the harmonic number. In the *Resistance* column, you enter the resistance. In the *Reactance* column, you enter the series reactance in ohms. Positive reactance adds inductance, and negative reactance adds capacitance. Multiple entries are allowed, so

© 1994-2020 All Rights Reserved.

different impedances can be set for each harmonic. The resistance or reactance can be left blank, which is interpreted as zero resistance or reactance.

Note: When a variable name is used to set the value of resistance or reactance, that variable must be added to ADE Explorer manually. The *Variables - Copy From Cellview* menu option will not enter the variable into the variable table in ADE Explorer. To add a variable manually:

1. Double-click on the last variable in the variables list. The *Editing Design Variables* window is displayed.
2. In the *Editing Design Variables* window, click *Next*. The *Name* and *Value (expr)* fields become editable.
3. Enter the variable name in the *Name* field, and the starting value in the *Value (expr)* field.
4. Click *OK* or *Apply*. The variable and value are entered into the table.

Note: Whenever you have an expression for one of the values, you must enclose the expression in parentheses. If this is not done, a simulation error is generated:

```
ERROR: Length of hvec is not equal to length of rvec multilies tones or length of
rvec is not equal to length of xvec.
```


© 1994-2020
All Rights Reserved.

Here is a comparison where the third harmonic is set normally in the top trace, and to 1 ohm in series with +j1mohm in the bottom trace. The lower impedance for the third harmonic in the bottom trace lowers the amplitude of the third harmonic.

© 1994-2020
All Rights Reserved.

Two Input Frequencies

Now the harmonic balance *Choosing Analyses* form has two frequencies.

© 1994-2020 All Rights Reserved.

Also, the diamond cut is set with a very large *MaximOrder*. The reason for doing this is to see the harmonic numbers that correspond to the frequencies produced by the circuit. Each frequency has two harmonic numbers that indicate which harmonics are mixing with which to produce a specific output frequency. For example, take the 2.44GHz term. The harmonic number is 1 0. The frequency that is produced is 1*2.44G + 0*2.46G. Now look at the intermod at 2.42GHz. This index is 2 -1. The frequency is 2*2.44G - 1*2.46G. These are the harmonic numbers that can be set for the harmonic port. Wild cards (*) are not allowed in the harmonic number specification.

Note that if there were three input frequencies, there would be three numbers that specify every output frequency.

In this case, if the harmonic was specified as 3, there would be an error produced by Spectre because there is a single index in the harmonic, but there are 2 indices for all the harmonics of the existing simulation. The number of entries in the *Harmonic* field must match the number of input frequencies in the circuit.

The above example sets the impedance of the third-order intermods to be much lower than 50 ohms. If the source is in series, setting the resistance and/or reactance to a large number can block harmonics you don't want. If the port is in parallel like this one, setting small resistance and reactance values can short out an undesired harmonic. This can be used for debugging the circuit, or for setting values to maximize power-added efficiency. The

© 1994-2020 All Rights Reserved.

resistance or reactance can be left blank, which is interpreted as zero resistance or reactance.

The top trace above is from the normal port, and the bottom trace is from the harmonic port. The third order harmonics are much smaller because they are essentially shorted out.

Source type

The *Source type* parameter lets you select a wave shape for the port from the *Source type* cyclic field: *dc*, *pulse*, *exp*, *pwl*, *pwlz, sine, bit*, *prbs*, *oscmacro*, or blank. Each *Source type* has different parameter settings associated with it. You can define several different wave shapes and quickly switch between them without losing the wave shape settings. The wave shape settings are described in detail in the following sections:

- DC Parameters on page 1730
- Pulse Waveform Parameters on page 1731
- PWL Waveform Parameters on page 1745
- Sinusoidal Waveform Parameters on page 1768

© 1994-2020 All Rights Reserved.

- Exponential Waveform Parameters on page 1795,
- Bit Waveform Parameters on page 1797,
- PRBS Waveform Parameters on page 1806

The typical *Source types* used in SpectreRF analyses are: *dc*, *piecewise linear, pulsed*, and *sine*. For example, you can quickly switch from a sinusoid (for PSS analysis) to a DC level (for PAC analysis) by changing the *Source type* from *sine* to *dc*. The sinusoidal properties remain with the port, and can be changed when the *Source type* is returned to *sine*. This is true when switching between any of the sourcetypes. All the properties are retained for everything that you define.

When you set *Source type* to a blank value, the *port* acts as a resistive load.

DC Parameters

To generate a dc level from the *port* component, select *dc* in the *Source type* cyclic field. When the *Source type* is set to *dc*, the *dc* and *temperature effect* parameters are active.

The *dc* setting sets the DC level for all analyses.

DC voltage

The *DC voltage* parameter sets the port's DC level for all analysis. The value must be a real number. If you do not specify the DC value, the default value is 0 volts.

The *DC voltage* parameter specifies the DC voltage across the *port* when it is terminated in its reference resistance. In other words, the *DC voltage* of the internal voltage source is double the specified DC value, *dc*. Because all small signal analyses (AC, SP, XF, and Noise) use DC analysis results, the *DC voltage* level also affects the small-signal analyses.

© 1994-2020 All Rights Reserved.

Figure 8-3 Source type=dc in the Edit Object Properties form

The *Display small signal params, Display temperature params,* and *Display noise parameters* fields are discussed in Small-Signal Parameters on page 1858, "Temperature Effect Parameters" on page 1860, and "PRBS Waveform Parameters" on page 1806.

Pulse Waveform Parameters

The pulse waveform can generate a step, a single pulse, or a periodic pulse waveform.

To generate a pulse waveform from the *port* component, select *pulse* in the *Source type* cyclic field.

When you specify the voltage on a *port*, you are specifying the voltage when the port is properly terminated, and not the voltage on the internal voltage source. Therefore, the voltage on the internal source is set to twice the value specified on the *port*.

© 1994-2020 All Rights Reserved.

Figure 8-4 Source type=pulse in the Edit Object Properties form

Frequency name 1

The *Frequency name 1* parameter is listed in the *Fundamental Tones* section of the *Choosing Analyses* form for pss, qpss, and hb when *Names* is selected.

Important

If you have two different sources at the same frequency or at integer multiple of the same frequency, make sure that you use the same *Frequency name* 1 or 2 property for both sources. An example might be RF1.

© 1994-2020 All Rights Reserved.

Delay time

The *Delay time* parameter is the simulation time that the port stays at the zero value without becoming periodic. Generally, this should be set less than the period of the waveform. Longer times are allowed, however, this forces a transient analysis until the longest delay time before the large-signal analyses pss, qpss, and hb can start. If a 1GHz signal and a delay time of 1 second is in your circuit, a transient is forced for 1 billion cycles of the input before the pss, qpss, or hb can start. This effectively prevents them from starting because of the long simulation time in the transient analysis. This must be a real number.

Zero value

The *Zero value* property is the time-zero value of the pulse in Volts. The default is zero volts.

One value

The *One value* property specifies the voltage that the signal transitions to at time equals zero plus. The default is one volt, which is unlikely to be what you want. Always set this property.

Period of waveform

The *period* property specifies the period of the pulse. If this property is left blank, a single pulse is generated at the beginning of the transient simulation.

Rise time

The *Rise time* property specifies the zero to 100% risetime for the pulse. This is the time for the complete transition from the zero value to the one value.

Fall time

The *Fall time* property specifies the zero to 100% falltime for the pulse. This is the time for the complete transition from the one value to the zero value.

Pulse width

The *Pulse width* property specifies the time to remain at the one value. The default is infinity.

© 1994-2020 All Rights Reserved.

Pulse Waveform Examples

Generating a step from zero to 1.2 Volts

© 1994-2020
All Rights Reserved.

1. Set the *Zero value* field to 0 volts.
2. Set the *One value* field to 1.2 volts.
3. Set the risetime to a time that is appropriate for your circuit. A suggested starting value is about $\frac{1}{10 \times CircuitBandwidth}$.

   The default is the falltime. If the falltime is not given, the default is $\frac{Period}{100}$. If the period is not given, then the default is $\frac{SimulationTime}{100}$.

   The waveform is shown below.

© 1994-2020
All Rights Reserved.

4. Setting a *Delay time* to 1n shifts the waveform to the right.

© 1994-2020
All Rights Reserved.

The waveform is shown below.

Specifying a startup bump in a supply voltage

This is commonly used to start oscillators, and is usually done using a `vsource`.

1. Set the *Zero value* to about 80% of the supply voltage.
2. Set the *One value* to the supply voltage.

© 1994-2020 All Rights Reserved.

3. Set the *Rise time* as appropriate for the system you are simulating.

© 1994-2020
All Rights Reserved.

4. The waveform is shown below.

© 1994-2020
All Rights Reserved.

Specifying a single pulse at startup, then zero after that.

This is usually used with the ipulse or isource components to start an oscillator.

© 1994-2020
All Rights Reserved.

1. Set the *Zero value* to zero.
2. Set the *One value* large enough to get things started.
3. Set the rise and fall times to about one tenth of the period of the oscillator.
4. Set the *Pulse width* to about three tenths of the period.
5. Leave the *Period of waveform* field blank.
6. The waveform is shown below.

To generate a pulse with an arbitrary duty cycle and differing rise and fall times:

1. Set the *Zero value* and *One value* fields to the levels needed for your system.
2. Set the *Rise time* and *Fall time fields* to approximate the real input source. The values can be different.
3. Set the *Period of waveform* field to the value needed for your frequency.

© 1994-2020 All Rights Reserved.

4. Set the *Pulse width* between 0.05 to about 0.95 times the period.

© 1994-2020
All Rights Reserved.

5. The waveform is shown below.

To generate a 50% duty cycle square wave:

1. Set the *Zero value* and *One value* fields as appropriate for your system.
2. Set the *Rise time* and *Fall time fields* as appropriate for your system and make them equal.
3. Set the *Period of waveform field* to that needed to generate your input frequency.

© 1994-2020 All Rights Reserved.

4. Set the *Pulse width* to half the period minus the risetime.

© 1994-2020
All Rights Reserved.

5. The waveform is shown below.

PWL Waveform Parameters

Piecewise linear waveforms allow an arbitrary input waveform to be generated. The input can either be a file that contains time and voltage pairs, or you can enter the time-voltage pairs directly in the PWL source properties form. Remember that the voltages you enter in the piecewise linear file assume that the port is properly terminated. The internal voltage source gets set to double the value specified in the piecewise linear voltage specifications.

Important

If the period property is set, the source repeats the waveform between zero and period over and over again. If the period property is not set and the transient analysis goes past the last point in the PWL file, the last voltage value in the file is held until the end of the simulation. If the period property is not set, the signal does not become periodic until after the time of the last point in the PWL file, where the source output becomes a DC level. If the file has a stopping point that has a large time, then the periodic analyses pss, qpss, and hb run a transient until the longest time in any

© 1994-2020 All Rights Reserved.

input PWL file has been reached before they go into the steady-state analyses, and these analyses use the DC level that is present at the end of the PWL file as the input signal.

Important

If you want a periodic piecewise linear waveform for pss, qpss, and hb, make sure that the period property is set.

To generate a piecewise linear waveform from the *port* component, select *pwl* in the *Source type* cyclic field. This sets the *Source type* CDF parameter to *pwl* and displays additional fields for the PWL CDF parameter settings.

Important

The advantage of entering a square or pulse wave using the pwl source is that the starting value can be at the center of the waveform. In the pulse waveform, the signal starts at the zero value, and because of this, in some cases, the tstab time must be extended to allow the waveform to settle. Starting the waveform in the middle can drastically reduce tstab and/or dramatically improve convergence for systems that have square or pulse waves and relatively long settling times for the nodes with the square or pulse wave signal.

When you specify the voltage on a *port*, you are specifying the voltage when the port is properly terminated, and not the voltage on the internal voltage source. Therefore, the voltage on the internal source is set to twice the value specified on the *port*.

© 1994-2020 All Rights Reserved.

Figure 8-5 Source type=pwl in the Edit Object Properties form

Waveform Entry Method

With the *Waveform Entry Method* buttons, select how you enter piecewise-linear data,

- By specifying a *File name*.
- By entering a series of *Voltage/Time points* directly in the properties list for the port.

© 1994-2020 All Rights Reserved.

File name

When you select *File* as the *Waveform Entry Method*, type the name of the file containing your piece wise-linear data in the *File name* field. The file name must be a string. There is no default. In ADE Explorer, define the directory that contains the file by selecting *Setup - Simulation Files*, and entering either the rooted path (starts with slash (/)) or relative path (starts with dot (.)) in the *Include Paths* field. Multiple directories can be specified in the *Include Paths* fields.

In your file, list the piecewise-linear data in the form of time-value pairs. Enter one pair per line with a space or tab between the time and voltage values. The numbers in the file must use exponential notation, for example, 1.04e-3. You can also use SI scale factors (p, n, u, m, k, M, G, etc.).

Number of PWL/Time pairs

When you select *Voltage/Time points* as the *Waveform Entry Method*, the *Number of PWL-Time pairs* field (tvpairs) form opens. Enter the number of time-value pairs you plan to enter. The form expands to let you enter the designated number of *Time* and *Voltage* values. Units are seconds and volts. The default is 0 and the maximum value is 50.

In this example, the number of voltage-time pairs is 6.

© 1994-2020 All Rights Reserved.

Figure 8-6 Waveform Entry Method=Voltage/Time points

© 1994-2020
All Rights Reserved.

The waveform that is produced is shown below.

The *DC offset, Amplitude scale factor,* and *Time scale factor* parameter fields let you quickly adjust the amplitude, frequency, and offset of your piecewise-linear data pairs without editing each individual time-value pair in the *PWL* waveform.

DC offset

DC offset for the *PWL* waveform. The default is zero volts.

Important

Note that for PWL waveforms, the DC Voltage term in the port only applies for the DC based analyses and not for the PWL waveform itself. To shift the DC level of the PWL waveform in the large-signal analyses (tran, envlp, pss, qpss, and hb) use the DC offset property. When the DC voltage property and the DC offset property are both set, the actual offset value used in the large signal analyses is defined by the DC offset property by itself, and not the DC voltage plus the DC offset property.

© 1994-2020 All Rights Reserved.

To show the effect of setting both the DC voltage and DC offset properties, in the following example, the DC Voltage and the DC offset are both set to 0.1.

© 1994-2020
All Rights Reserved.

Note that the signal only observes the DC offset in the transient.

© 1994-2020
All Rights Reserved.

Amplitude scale factor

Amplitude scale factor (*scale*) for the *PWL* waveform. The default value is 1. The example below shows a scale factor of 3.

© 1994-2020
All Rights Reserved.

Note that the waveform is multiplied by three in amplitude.

© 1994-2020
All Rights Reserved.

Time scale factor

Time scale factor is a multiplier for the times given for *PWL* waveform. Default is 1. Setting the time scale factor greater than 1 increases the times in the file, and reduces the frequency.

© 1994-2020
All Rights Reserved.

Note that setting the time scale factor to 0.5 multiplies all the times including the time in the period property.

Cosine Filter

The raised cosine filter is used when the PWL source is used to define digital modulation. Values are `none` (default), and `nrc`. `nrc` specifies a raised cosine filter to be used on the output waveform of the PWL source. This can be used to lower the numerical noise floor of

© 1994-2020 All Rights Reserved.

the modulation specified in the I and Q modulating file. When `nrc` is selected, the PWL file must have evenly spaced points with timepoints at 1/(2*Bandwidth of the filter).

© 1994-2020
All Rights Reserved.

Rolloff factor

When `nrc` is selected, the *Rolloff factor* (*rolloff*) of the filter can be specified. The default value is `0.2`, which is reasonable for simulations. If you consider a continuous waveform, changing the rolloff factor has an effect on the ringing that is produced in the time domain. Smaller rolloff factors produce more ringing in the continuous waveform. Envelope does not simulate the continuous waveform. Instead, envelope samples the waveform at the time interval in the I and Q modulation file. This delta-time in the I and Q file is always related to the modulation frequency, and it also determines the bandwidth of the cosine filter as described below. The effect of this is that the sampling occurs at the times where the ringing of the filter is zero. Therefore, the cosine filter does not change the spectral content calculated in envelope. It just reduces the numerical noise floor of the input signal.

Bandwidth

The *Bandwidth* (*pwlbandwidth*) of the Cosine Filter. If you want to only allow in-band modulation, this is the channel bandwidth. If you want to allow an accurate ACPR measurement, set the bandwidth to three times the channel bandwidth. If you want to measure two adjacent channels for ACPR, set the bandwidth to five times the channel bandwidth. Note that the I and Q data must have properly spaced time intervals of $\frac{1}{2 \times Bandwidth}$ for this capability to be used.

Note: The *Rolloff factor* and *Bandwidth* options are available only when `nrc` is selected in the *Cosine Filter* drop-down list.

Desired rms value

Desired rms voltage for the PWL waveform. Note that this will be set based on the datapoints in the PWL file or in the manually defined points entered directly into the properties form only. This has implications for periodic piecewise linear or if you go beyond the last point in the file in the transient analysis.

If the signal is defined for only part of the period or if you go beyond the last point in the pwl description, the rms calculation will not take into account the fact that the signal remains at the last voltage after the last timepoint in the pwl file. If you want this to be taken into account, define a point in the pwl description at the stop time of the transient or at the end of the period.

© 1994-2020 All Rights Reserved.

Consider an example in which a 1GHz signal is being generated, with points for the first period of 1GHz defined. No points after 1nsec exist in the PWL description. Also note that the desired RMS value is set to 1.5 (volts).

© 1994-2020
All Rights Reserved.

Now the transient result is plotted.

Vertical markers have been placed at the midpoint of the high and low states. The voltage is slightly greater than 1.5 because of the finite rise and fall times in the PWL description.

© 1994-2020
All Rights Reserved.

Now, the period is set to 2nsec. This will make the waveform periodic at 500MHz.

© 1994-2020
All Rights Reserved.

Now the transient result is plotted. Note that the high and low values are the same as before.

Now the transient waveform is sent to the calculator, and the actual rms value is calculated.

Instead of producing 1.5 volts as specified, only 1.061 volts is produced.

© 1994-2020
All Rights Reserved.

Now add another datapoint in the PWL description at 2nsec, and zero volts. Now the period and the time interval that defines the PWL points exactly agree.

© 1994-2020
All Rights Reserved.

Now the transient result is plotted.

Note that the positive and negative voltages have changed. This is because the PWL description completely defines the waveform during the entire first period of the waveform.

Now the signal is set to the calculator, and the rms is calculated. It is appended to the last table.

Now the RMS value is exactly 1.5 volts.

© 1994-2020
All Rights Reserved.

Now leave the last timepoint at 2nsec, and reduce the period to 1nsec.

© 1994-2020
All Rights Reserved.

Now, the transient result is plotted. The positive and negative voltages still reflect the rms value for the 2nsec interval defined by the timepoints themselves. It has not truncated the timepoints to just the first period. It is also not periodic. When the period is smaller than the time of the last timepoint in the PWL description, Spectre assumes that the period is in error, and ignores the property.

Tip

If you need to set an rms voltage for a periodic piecewise linear signal, make sure that the timepoints in the piecewise linear description define timepoints over exactly the entire first period of the waveform.

Do not set *Desired rms value* and *Power of PWL waveform* at the same time. If both are specified, first the amplitude is calculated based on the Power of PWL waveform property, and then that amplitude is multiplied by the *Desired rms value* setting.

Delay time

The *Delay time* parameter sets the time that the signal remains at the first point in the file before the pwl points begin to take effect. Generally, this should be set less than the period of the waveform. Longer times are allowed, however, this forces a transient analysis until the

© 1994-2020 All Rights Reserved.

longest delay time before the periodic large-signal analyses pss, qpss, and hb can start. If a 1GHz signal and a delay time of 1 second is in your circuit, a transient is forced for 1 billion cycles of the input before the pss, qpss, or hb can start. This effectively prevents them from starting because of the long simulation time in the transient analysis.

Breakpoints

Possible values are *no*, *yes*, or blank. If you set *Breakpoints* to *yes*, you force SpectreRF to place time points at each point specified in a *PWL* waveform during a transient analysis. This can be very expensive for waveforms with many points. If you set *Breakpoints* to *no*, SpectreRF inspects the waveform, looking for abrupt changes, and forces time points only at those changes. If you set *Source type = pwl* and set *Breakpoints* to blank, the default is *yes* if the number of points you specify is less than 20.

Period

The *PWL* waveform is periodic if you specify *Period* in seconds.

If the value of the waveform you specify is not exactly the same at both its beginning and its end, then you must provide a non-zero value for *Transition Width*.

Power of PWL waveform

This is the power in dBm for a 50 ohm resistor for the output of the PWL source. See Desired rms value on page 1758 for important information about how to define timepoints in the PWL input points. Do not set *Desired rms value* and *Power of PWL waveform* at the same time. If both are specified, first the amplitude is calculated based on the *Power of PWL waveform* property, and then that amplitude is multiplied by the *Desired rms value* setting.

Transition Width

Transition width (*twidth*) is used when making *PWL* waveforms periodic and the ending value of the PWL file does not equal the beginning value. The default is the `PWL period/1000`.

Before repeating, the waveform changes linearly in an interval of *Transition Width* from its value at *(Period – Transition Width)* to its value at the beginning of the waveform. Thus, the *Transition Width* must always be much less than the *Period*.

© 1994-2020 All Rights Reserved.

Sinusoidal Waveform Parameters

The *port* component can generate up to two sinusoids simultaneously. They are denoted as 1 and 2. You can set the amplitude, frequency, and phase for both individually. The amplitude can be set to either a voltage or a power level. When you set a power level, the assumption is that the port is perfectly matched. The source that is internal to the port gets double the amplitude specified by the power in dBm.You can also specify sinusoidal AM or FM modulation of sinusoid 1. Sinusoid 2 cannot be modulated.

To generate sinusoidal waveforms, set the *Source type* in the properties list to *sine*.

The first sinusoid is described by the parameters *Frequency name 1, Frequency 1, Amplitude 1 (Vpk)*, *Amplitude 1 (dBm)*, *Phase for Sinusoid 1*, *Sine DC level*, and by AM or FM modulation terms.

© 1994-2020 All Rights Reserved.

Figure 8-7 Source type=sine in the Edit Object Properties form

Frequency name 1

This property names the fundamental tones of sinusoid 1. After you save the schematic, the names you assign appear in the *Fundamental Tones* list box on the *Choosing Analyses* form.

© 1994-2020 All Rights Reserved.

Important

When you have multiple ports that are connected to the circuit at different places, but have the same frequency, or integer multiples of the same frequency, make sure that the *Frequency name 1* property is the same for every instance. An example might be RF1. For harmonic balance, when names is selected in the *Choosing Analyses* form, and for qpss, the frequency name property specifies that all the signals with the same name should be treated as one tone in the simulation. This can save a lot of time when you run those analyses.

Frequency 1 is the frequency of the first sinusoidal waveform (carrier frequency). You typically use unmodulated signals in SpectreRF analyses. The value must be a real number. The default frequency is zero, which causes no signal to be generated. Always set this property.

Amplitude 1 (Vpk)

The peak amplitude of the first sinusoidal waveform that you generate. The value specified is the voltage delivered into a matched load. You can select either *Amplitude 1 (Vpk)* or *Amplitude 1 (dBm)*, but not both. If *Amplitude 1 (Vpk)* has a value, the *Amplitude 1 (dBm)* field is grayed out. The value must be a real number. The default is 1 volt.

When you specify the voltage on a *port*, you are specifying the voltage when the port is properly terminated, and not the voltage on the internal voltage source. Thus, the voltage on the internal source is set to twice the value specified on the *port*.

Amplitude 1 (dBm)

Amplitude 1 (dBm) is the amplitude of the first sinusoidal waveform, in dBm.The resistance for the dBm calculation is taken from the resistance of the port. The value specified is the power delivered into a matched load. You can select either *Amplitude 1 (Vpk)* or *Amplitude 1 (dBm)*, but not both. If *Amplitude 1 (dBm)* has a value, the *Amplitude 1 (Vpk)* field is grayed out. The value must be a real number.

Phase for Sinusoid 1

The phase at the specified *Delay time*. To achieve a specified phase and still remain continuous, the sinusoidal waveform might start before the given *Delay time*. For example, if you want to generate a cosine wave, set this parameter to 90°. The value must be a real number. The default is 0 degrees.

© 1994-2020 All Rights Reserved.

Important

For circuits that have long settling times for the nodes connected to the sinusoidal signal, it can save a lot of time and/or dramatically improve convergence if you set phases other than zero or 180 degrees by using the delay time parameter instead of the *Phase* parameter. Imagine a need to start at a 90 degree phase shift. If the *Phase for Sinusoid 1* property is set to 90, the waveform that is produced starts at the positive peak, If instead, the delay time was set to 3/4ths of the period and the *Phase for Sinusoid 1* property was set to zero, instead of starting at the peak and needing to settle to near the middle value in the tstab interval, the signal itself starts at the center value, so the circuit has to settle a much smaller amount.

Sine DC level

Sets the DC level in volts for sinusoidal waveforms in large-signal analyses. This parameter is used when the sinusoid has a different average level than the one specified for the DC analyses. If not specified, the average value of the sinusoid is the same as that of the DC level of the source.

Important

When this property is specified, it is used directly. The DC level of the sinusoid is not the DC voltage plus the Sine DC level. It is just the Sine DC level.

Modulation Parameters

Display Modulation Parameters

Important

The check box is a display feature in the form only. If there are properties that are set and the display check box is not selected, although you will not see the parameters in the form, these parameters will still be netlisted, and will still have an effect on the simulation.

When selected, the form expands and the following modulation parameters are displayed: *FM modulation index, FM modulation freq, AM modulation index, AM modulation freq, AM modulation phase,* and *Damping factor*.

© 1994-2020 All Rights Reserved.

Important

These parameters only affect the sinusoidal waveform.

Important

Only the first `sinusoid` can be modulated.

Figure 8-8 Display modulation params

AM Modulation (Background Information)

The amplitude modulation for the sinusoidal case is defined as

$$V(t) = (A \bullet \sin(\omega_C t)) + \frac{AM}{2}[\sin((\omega_C + \omega_{Mod})t + \phi) + \sin((\omega_C - \omega_{Mod})t - \phi)]$$

where

- $V(t)$ is the amplitude modulation waveform
- ω_C is the carrier frequency
- AM is the AM modulation index
- ω_{Mod} is the modulation frequency
- ϕ is the *AM modulation phase*

© 1994-2020 All Rights Reserved.

The amplitude modulation parameters affect only the first sinusoid generated by *port*. They have no effect on the second sinusoid.

AM modulation Index

This property specifies the amplitude of the modulation signal. When the AM modulation index is 1.0, the signal is 100% modulated. When the AM modulation index is 0.5, the signal is 50% modulated.

AM Modulation Frequency

This property specifies the frequency of the sine wave that is used to modulate the signal that is specified in the *Frequency 1* property. The AM modulation frequency should be much lower than the *Frequency1* property.

AM Modulation Phase

This specifies the initial phase of the AM modulation sinusoid. The default is zero degrees.

© 1994-2020 All Rights Reserved.

The *Edit Object Properties* form for a 100% modulated signal is shown below.

© 1994-2020
All Rights Reserved.

The output waveform is shown below.

FM Modulation (Background Information)

The frequency modulation for the sinusoidal case is defined as:

$V_{FM}(t) = A \sin(2\pi f_c t + \beta \sin(2\pi f_m t) + \phi)$

where

- `A` is the amplitude of sinusoid 1
- β is the FM modulation index
- $\sin(2\pi f_m t)$ is the modulation signal
- f_c is the carrier frequency
- ϕ is the *phase for sinusoid 1*

The frequency modulation parameters affect only the first sinusoid generated by *port*. They have no effect on the second sinusoid.

© 1994-2020 All Rights Reserved.

FM modulation frequency

FM modulation frequency for the sinusoidal waveform (`fm` in the previous equation).

FM modulation index

FM index of modulation for the sinusoidal waveform. It is the ratio of the peak frequency deviation divided by the modulation frequency (β in the above equations).

$\beta = \Delta f / f_m$

When the modulation index is 1.0, the deviation frequency from the carrier frequency is equal to the modulation frequency.

The property list for an FM signal with a modulation index of 100% is show below.

© 1994-2020 All Rights Reserved.

© 1994-2020
All Rights Reserved.

The transient waveform goes from zero frequency to twice the center frequency. This is an extremely large modulation index. Zero frequency is produced at the center of the below waveform.

© 1994-2020
All Rights Reserved.

When zoomed in near the beginning, the frequency is nearly double the carrier frequency.

Number of FM Files

The first sinusoid can also be phase modulated with a pair of PWL files that define the desired modulation.

- *none* specifies that the FM signal should be modulated with a sinusoid.

  In this case, the FM signal from the source will be represented as:

  $V_{FM}(t) = A \sin(2\pi f_c t + \beta \sin(2\pi f_m t))$

- When *one* is selected, a *Name of FM File1* field appears. Provide the filename of the PWL file that is to be used for the frequency modulation.

  In this case, the FM signal from the source will be represented as:

  $$V_{FM}(t) = A \sin(2\pi f_c t + \beta \int_0^t \delta\omega(\tau) d\tau)$$

© 1994-2020 All Rights Reserved.

This is shown in the *Edit Object Properties* form, as shown below.

© 1994-2020
All Rights Reserved.

- When *two* is selected, the *Name of FM File1 and Name of FM file2* fields appear. Provide the filename of the PWL file that is to be used for the phase modulation.

  In this case, the FM signal from the source will be represented as:

  $$V_{FM}(t) = A(t)\sin(2\pi f_c t + \beta\theta(t))$$

  Here,

  β is the frequency modulation index

  $$A(t) = \sqrt{(I(t))^2 + (Q(t))^2}$$

  where `I(t)` is the waveform specified in the *Name of FM File1* pwl file, and `Q(t)` is

  $$\theta(t) = \arctan\left(\frac{Q(t)}{I(t)}\right)$$

© 1994-2020 All Rights Reserved.

waveform specified in the *Name of FM File2* pwl file. This is shown below.

© 1994-2020 All Rights Reserved.

© 1994-2020
All Rights Reserved.

Modulating a carrier with a digital modulation I and Q file

This capability provides exactly the same capability as the *ACPR_Source* in rfLib. The I and Q PWL files provide the information to be modulated, and the source acts as an ideal modulator.

IQ modulation I File

This parameter should be set to the filename to the PWL file that contains the digital modulation 'I' information. This is a file with two columns. The left column is the time, and the right column is the voltage. Usually, the file is generated with a constant delta-T from entry to entry in the file. Generally, this file is provided by the system designer.

IQ modulation Q File

This parameter should be set to the filename to the PWL file that contains the digital modulation 'Q' information.This is a file with two columns. The left column is the time, and the right column is the voltage. Usually, the file is generated with a constant delta-T from entry to entry in the file. Generally, this file is provided by the system designer.

Cosine Filter

The raised cosine filter is used when the PWL source is used to define digital modulation. Values are `none` (default), and `nrc`. `nrc` specifies a raised cosine filter to be used on the output waveform of the PWL source. This can be used to lower the numerical noise floor of the modulation specified in the I and Q modulating file. When `nrc` is selected, the PWL file must have evenly spaced points with timepoints at 1/(2*Bandwidth of the filter).

Rolloff factor

When `nrc` is selected, the *Rolloff factor* (*rolloff*) of the filter can be specified. The default value is `0.2`, which is reasonable for simulations. If you consider a continuous waveform, changing the rolloff factor has an effect on the ringing that is produced in the time domain. Smaller rolloff factors produce more ringing in the continuous waveform. Envelope does not simulate the continuous waveform. Instead, envelope samples the waveform at the time interval in the I and Q modulation file. This delta-Time in the I and Q file is always related to the modulation frequency, and it also determines the bandwidth of the cosine filter as described below. The effect of this is that the sampling occurs at the times where the ringing of the filter is zero, thus the cosine filter does not change the spectral content calculated in envelope. It just reduces the numerical noise floor of the input signal.

© 1994-2020 All Rights Reserved.

Bandwidth

The *Bandwidth* (*pwlbandwidth*) of the Cosine Filter. If you want to only allow in-band modulation, this is the channel bandwidth. If you want to allow an accurate ACPR measurement, set the bandwidth to three times the channel bandwidth. If you want to measure two adjacent channels for ACPR, set the bandwidth to five times the channel bandwidth. Note that the I and Q data must have properly spaced time intervals of $\frac{1}{2 \times Bandwidth}$ for this capability to be used.

Note: The *Rolloff factor* and *Bandwidth* options are available only when `nrc` is selected in the *Cosine Filter* drop-down list box.

The *Rolloff factor* and *Bandwidth* options are available only when `nrc` is selected in the Cosine Filter.

Damping factor

This property is available in the `vsin` and `isin` components only.

Damping factor for the sinusoidal waveform. 1/*Damping factor* specifies the time it takes to change the amplitude by a factor of e. For example, consider the following damped sinusoid:

$$v(t) = A\, e^{-\sigma t} \sin(2\pi f t + \phi)$$

where

σ = *Damping factor*, `A` is the amplitude of sinusoid 1, and ϕ is the *Phase for sinusoid 1*

- If $\sigma = 0$, the waveform is a pure sinusoid (steady state).
- If $\sigma < 0$, the waveform exhibits growing amplitude.
- If $\sigma > 0$, the waveform exhibits decaying amplitude.
- The value must be a real number. The default is 0 and the unit is 1/seconds.

© 1994-2020 All Rights Reserved.

For the vsin component, *am* example property list is shown below.

The amplitude is 1 volt. The initial phase of 90 degrees starts the sine wave at a peak so that the amplitude at 1/100M seconds or 10nsec will be at a peak that can be measured with a

© 1994-2020 All Rights Reserved.

marker in the waveform tool. The transient analysis has strobeperiod set so a datapoint at exactly 10nSec is produced in the output. With a positive damping factor, the waveform decays, and the amplitude is 1/e at 10nsec.

© 1994-2020
All Rights Reserved.

When the damping factor is negative, the amplitude is e (2.7182...) at 10 nsec.

Display second sinusoid

Important

The check box is a display feature in the form only. If there are properties that are set and the display check box is not selected, although you will not see the parameters in the form, these parameters will still be netlisted, and will still have an effect on the simulation.

Displays the CDF parameters for the second sinusoid in the *Edit Object Properties* and *Add Instance* forms. When selected, the form expands to show the following CDF parameters: *Frequency name 2, Frequency 2, Amplitude 2 (Vpk), Amplitude 2 (dBm),* and *Phase for Sinusoid 2*.

Important

The second sinusoid cannot be modulated.

© 1994-2020 All Rights Reserved.

Figure 8-9 Display second sinusoid

Frequency name 2

Name for the second sinusoid. After you save the schematic, the name you assign is listed in the *Fundamental Tones* section of the *Choosing Analyses* form for pss, qpss, and hb when names is selected.

Important

If you have two different sources at the same frequency or at integer multiple of the same frequency, make sure that you use the same frequency name 1 or 2 property for both sources. An example might be RF1.

Frequency 2 is the frequency of the second sinusoidal waveform in Hertz. There is no default value.

Amplitude 2 (Vpk)

Peak amplitude of the second sinusoidal waveform. The value specified is the voltage delivered into a matched load. You can select either *Amplitude 2 (Vpk)* or *Amplitude 2 (dBm)*, but not both. If *Amplitude 2 (Vpk)* has a value, the *Amplitude 2 (dBm)* field is grayed out. The value must be a real number. The default is 1 volt.

When you specify the voltage on a *port*, you are specifying the voltage when the port is properly terminated, and not the voltage on the internal voltage source. Thus, the voltage on the internal source is set to twice the value specified on the *port*.

Amplitude 2 (dBm)

Amplitude 2 (dBm) is the amplitude of the second sinusoidal waveform, in dBm. The resistance for the dBm calculation is taken from the resistance parameter of the port. the value specified is the power delivered into a matched load. You can select either *Amplitude*

© 1994-2020 All Rights Reserved.

2 (Vpk) or *Amplitude 2 (dBm),* but not both. If *Amplitude 2 (dBm)* has a value, the *Amplitude 2 (Vpk)* field is grayed out. The value must be a real number. Units: dBm

Phase for Sinusoid 2

The phase at the specified *Delay time* for the second sinusoid. To achieve specified phase while still remaining continuous, the sinusoidal waveform might start before the given *Delay time*. The value must be a real number. Default: 0 Units: degrees.

Display multi sinusoid

Important

The check box is a display feature in the form only. If there are properties that are set and the display check box is not selected, although you will not see the parameters in the form, these parameters will still be netlisted, and will still have an effect on the simulation.

Multi sinusoid is a way of generating up to nine unmodulated signals at the same time from the port component. It works in all the large-signal analyses, and is usually used with harmonic balance or qpss-harmonic balance for the simulation of systems with multiple carriers.

When *Display multi sinusoid* is selected, the properties form expands to show more properties. The *Number of Frequencies* field specifies the number of carriers you want to generate. Up to nine carriers can be generated by the port component. For each one of the frequencies, a set of parameters patterned after: *Sinusoid Frequency 1, Sinusoid Ampl 1 (Vpk), Sinusoid Ampl 1 (dBm),* S*inusoid Phase 1, Sinusoid Maxharm 1* is displayed for each tone you want to generate.

When *Display multi sinusoid* is selected, the frequency specified in the *Frequency1* parameter should be the middle frequency of the group of frequencies you want to generate. It must be one of the frequencies that could actually be generated by the port component. If four frequencies are to be generated, then the *Frequency1* property could be either the second or third frequency in the series.

Consider an example where 2.43GHz, 2.44GHz, and 2.46GHz need to be generated for a composite triple beat measurement. The actual center frequency is 2.445GHz. Since this is not one of the frequencies in the series that is actually generated, it cannot be picked. Pick one of the frequencies that could be generated on either side of the actual center frequency. Here, the center frequency could be either 2.44GHz or 2.45GHz. Even though 2.45GHz is not

© 1994-2020 All Rights Reserved.

being generated in this example, it is one of the frequencies that could be generated by the series, and thus it is eligible to be selected.

The frequency specified in the *Frequency 2* field should be the spacing frequency. All the frequencies specified in the multi-frequency section need to be separated by the spacing frequency. In the example above, the spacing frequency is 10MHz.

Up to nine frequencies can be generated in one port. When this feature is used with HB or QPSS, only specify the *Frequency1* and *Frequency2* frequencies in the frequencies section. The number of harmonics should be set to about two times the total number of tones that are generated in the multi-frequency section.

© 1994-2020 All Rights Reserved.

Figure 8-10 Display multi sinusoid

Number of Frequencies

Number of sinusoid frequencies to be specified.

© 1994-2020 All Rights Reserved.

Sinusoid Frequency 1

The frequency of the first sinusoidal waveform (carrier frequency) in hertz.

Sinusoid Ampl 1 (Vpk)

The peak amplitude of the sinusoidal waveform that you generate in volts. When you specify the voltage on a *port*, you are specifying the voltage when the port is properly terminated, and not the voltage on the internal voltage source. Thus, the voltage on the internal source is set to twice the value specified on the *port*.

Sinusoid Ampl 1 (dBm)

The amplitude of the first sinusoidal waveform when specified in dBm.

Sinusoid Phase 1

The phase at the specified delay time. To achieve a specified phase and still remain continuous, the sinusoidal waveform might start before the given delay time. For example, to generate a cosine wave, set this parameter to pi/2. The units are radians.

Sinusoid Maxharm 1

Do not set this parameter. Harmonics are usually set in the hb or qpss *Choosing Analyses* form.

© 1994-2020 All Rights Reserved.

Below is a plot of the output when the variable `prf` is set to -40 (dBm).

The amplitudes of the tones at 2.4G, 2.41G, 2.42G and so on are at the numerical noise floor of the simulation.

When these three frequencies are applied to a real system, the nonlinearity of the system produces mixing products. For example, 2.43G mixes with 2.44G to produce 10M. 2.44G mixes with 2.46G to produce 20M. 2.43G mixes with 2.46G to produce 30M. 2.43G, 2.44G, and 2.46G are first order terms, so 10M, 20M, and 30M are second order terms.

Now the 10M, 20M, and 30M terms can mix with 2.43G, 2.44G, and 2.46G. For example, 2.44G mixes with 10M to produce 2.45G. 2.44G is a first order term, and 10M is a second order term, so 2.45G is a third order term. 2.43G can mix with 20M to add an additional third

© 1994-2020 All Rights Reserved.

order term at 2.45G. This leads to many harmonics that are actually produced, as shown below. The vertical axes have the same scale.

Exponential Waveform Parameters

To generate an exponential waveform from the `port` component, set the CDF parameter *Source type=exp*, as shown in the next figure.

© 1994-2020 All Rights Reserved.

Figure 8-11 Source type=exp in the Edit Object Properties form

The exponential waveform can generate one exponential pulse, and cannot generate a periodic signal. This it is not usable with SpectreRF.

When you specify the voltage on a *port*, you are specifying the voltage when the port is properly terminated, and not the voltage on the internal voltage source. Thus, the voltage on the internal source is set to twice the value specified on the *port*.

Zero value

The *Zero value* property is the time-zero value of the exponential waveform in volts.

One value

The *One value* property specifies the voltage that the signal transitions to at time equals zero plus. The default is one volt, which is unlikely to be what you want. Always set this property.

Rise time start

This property specifies the simulation time in seconds to begin the transition from the zero to the one value. The default is zero seconds.

© 1994-2020 All Rights Reserved.

Rise time constant

The *Rise time constant* specifies the time for the exponential waveform to reach 63 percent of the way from the zero value to the one value. The default is 0.075 percent of the stop time in the transient analysis.

Fall time start

The *Fall time start* specifies the simulation time to begin the transition from the one state to the zero state. The default is one percent of the transient analysis stop time.

Fall time constant

The *Fall time constant* specifies the simulation time for the exponential waveform to reach 63 percent of the way from the one value to the zero value. The default is one percent of the stop time in the transient analysis.

Bit Waveform Parameters

To generate bit sequence/string from the port component, select bit in the *Source Type* cyclic field. The bit source has four states: `1`, `0`, `m`, and `z`, which represent the high, low, middle voltage/current and high impedance state respectively. It allows patterns defining a sequence of bits. When the *m* state is specified, the output voltage is set halfway between the zero and one voltage.

© 1994-2020 All Rights Reserved.

Figure 8-12 Source type=bit in the Edit Object Properties form

Delay Time

The *Delay time* property specifies the time the first bit in the string should be held before the first bit is generated.

Note: When the delay time property is set, the time of the first transition is at the delay time plus the period of the first bit, which is set by the *Period of waveform* property.

Zero Value

The *Zero value* property sets the voltage when the pattern specifies a zero state. The pattern will be defined shortly.

Important

For the zero and one value, the port is assumed to be terminated with a matching register. The internal voltage source is doubled.

If you intend to use the z state of the bit source, do not add a matching resistor across the port terminals, since this resistor will totally negate the high impedance state. Make

© 1994-2020 All Rights Reserved.

sure that you halve the voltage values you want for the zero and one states when you use the port element to generate a bit pattern.

One Value

The *One value* property sets the voltage of the port when the pattern specifies a one.

Period of Waveform

The bit waveform type generates a bit stream where the period defines the period of one of the bits in the stream. For example, if the period property is set to 1n, then every nanosecond the bit switches to the next bit in the sequence.

Rise Time

The *Rise time* property specifies the transition time from the zero state to the one state.

Fall Time

The *Fall time* property specifies the transition time from the one state to the zero state.

Pattern Parameter data

This is the bit string, which is a series of the four states; `1`, `0`, `m`, and `z`. An example is 101100 (not separated by spaces) which would generate a 1, then a zero, then 2 ones in a row, and then two zeros in a row.

rptstart

This specifies the starting bit number when repeating. Default value is `1`, which causes the first bit in the sequence to be the starting bit for the repeated pattern. The data repeats from the specified bit to the end of the bit string. The value of the parameter should be an integer from 1 to the length of the bit string.

rpttimes

This specifies the number of times the pattern should be repeated after the first sequence is generated. The default value is `0`. which specifies that the pattern should not be repeated. When its value is `negative`, the string repeats forever. If a finite number of repeats is

© 1994-2020 All Rights Reserved.

specified and the simulation time increases to a time after the last bit is generated, the output will maintain the state of the last bit in the sequence.

Bit Waveform Examples

Here is the circuit that is used for all the examples.

The port with the sourcetype set to bit is on the left. Then there is a 1MOhm resistor, and a DC voltage source set to 900mV.

© 1994-2020
All Rights Reserved.

Example 1

Here is a property list for the first example.

© 1994-2020
All Rights Reserved.

Here is the output signal which is *Out1* in the schematic.

The delay time is 5n, and the bit period is 1n. This causes the signal to start at the first bit which is 1, and hold it for 5nsec. Then, it produces the first bit, which is 1 for the next nanosecond. If you look closely, you can see that the middle of the transition occurs at exactly 6nsec. The start of the transition is half the falltime before. The sequence is 1010110000mmzzzzzzzzzz. You can see the 0, 1, 0 transitions at 7nsec, 8nsec, and 9nsec. Then you see the two 1 states from 9nsec to 11 nsec, and the four 0 states from 11nsec to 15nsec. The next two states are middle states at 0.6v from 15nsec to 17 nsec. Then the output goes to high impedance, where the 900mv DC level at the output is seen. Rpttimes is set to 1, which repeats the pattern once, and the starting bit is the first bit in the sequence, which is a 1. At the end of the 900mv section, the signal transitions to the 1 state, and the entire pattern is repeated again. After the second pattern is done, the last state which is a z state is held after that.

© 1994-2020 All Rights Reserved.

Example 2

Here is the property list for the second example.

© 1994-2020
All Rights Reserved.

Everything is the same as the first example except for the repeat start, which is set to 2. This causes the second bit in the sequence, which is a 0, to be the starting bit. At the end of the first pattern, the signal repeats from the second bit in the sequence.

© 1994-2020
All Rights Reserved.

Example 3

Here is the property list for the third example.

© 1994-2020
All Rights Reserved.

Here, there is no repeat.

PRBS Waveform Parameters

PRBS is an acronym for pseudo-random binary sequence. The PRBS source has three modes. It can be used to generate a maximum-length pseudo-random sequence. You can specify the beginning state and tap gains for a Fibonacci PRBS generator. A third mode allows reading an ascii file that describes the sequence of one and zero events to generate. To select this type of output, set the *Source Type* to *PRBS*.

© 1994-2020 All Rights Reserved.

For any mode, first you must define the output characteristics.

© 1994-2020
All Rights Reserved.

Delay Time

This specifies a time where the source stays at the zero value, and does nothing. The default is zero seconds. At the end of the delay time, the first bit will be produced.

Zero Value

This specifies the analog voltage to generate in the zero state. The default is zero volts. The port internally doubles the specified voltage as usual. To get the correct voltage when using the port, make sure that you add a matching resistor across the port terminals.

One Value

This specifies the analog voltage to generate in the one state. The default is one volt. The port internally doubles the specified voltage as usual. To get the correct voltage when using the port, make sure that you add a matching resistor across the port terminals.

Bit Period

This specifies the time of a single bit. It sets the trigger times between the zero and one state for the duration of a single bit. The transitions begin at the trigger event time. For example, if the *Bit Period* is 1n, then the first transition starts at 1nSec, the second transition starts at 2nSec, and so on.

Rise Time

This is the zero percent to 100 percent (by default) risetime for the transition from the zero state to the one state. The waveshape of the transition and the definition of risetime are settable.

Fall Time

This is the zero percent to 100 percent (by default) falltime for the transition from the one state to the zero state. The waveshape of the transition and the definition of risetime are settable.

Transition Reference

This specifies the thresholds for the rise time and fall time specification. This can be set to 0 to 100% (default), 10 to 90%, 20 to 80%, 30 to 70%, 40 to 60%, and *Specify*. When *Specify* is selected, the lower threshold must be entered. The upper threshold is calculated and is

© 1994-2020 All Rights Reserved.

equidistant from the midpoint as the lower threshold. For example, if 22% is set for the lower threshold, 78% will be the upper threshold.

© 1994-2020
All Rights Reserved.

Edge Type

The waveform shape of the transition defaults to linear, which generates a traditional pulse with straight edges. Selecting halfsine causes sinusoidal rounding of the edges.

PAM4 modulation

The current PRBS source has binary output. When *PAM4 modulation* is set to *PAM4*, the *prbs* source generates four levels of binary instead of two, which allows two bits per symbol instead of one.

PAM4 mapping

This option is available only when you select *PAM4* from the *PAM4 modulation* drop-down list. It specifies the mapping from a pair of bits to 4-level voltages. If you do not specify a value in this field, the default mapping is taken as 0132. This means:

00 generates 0V

01 generates 0.33V

11 generates 0.66V

© 1994-2020 All Rights Reserved.

10 generates 1.0V

© 1994-2020
All Rights Reserved.

Note that you are setting the bit period, not the symbol period. The symbol period is twice the bit period in PAM4.

The output shows four different voltages instead of two. The symbol period is 2ns.

LSFR Mode

This controls whether a maximal length sequence is generated, a file with a pattern of 1 and 0 states ia read and produced, or whether you specify the shift register, taps and starting the shift register state yourself.

Maximum Length Sequence LSFR Mode

The default is PN32, which generates a maximum length sequence (MLS) with a 32-bit shift register, and alternating 1 and zero states in the shift register as the starting state. To generate a different Maximum Length Sequence, change the seed parameter to define a different starting state, or define your own register, taps, and starting the state manually.

© 1994-2020 All Rights Reserved.

Maximum length sequences using 2 to 32 bit shift registers are available. The table below defines the taps for each length of the available shift registers.

Number of bits	Taps
2	2, 1, 0
3	3, 2, 0
4	4, 3, 0
5	5, 3, 0
6	6, 5, 0
7	7, 6, 0
8	8, 6, 5, 4, 0
9	9, 5, 0
10	10, 7, 0
11	11, 9, 0
12	12, 6, 4, 1, 0
13	13, 4, 3, 1, 0
14	14, 5, 3, 1, 0
15	15, 14, 0
16	16, 15, 13, 4, 0
17	17, 14, 0
18	18, 11, 0
19	19, 6, 2, 1, 0
20	20, 17, 0
21	21, 19, 0
22	22, 21, 0
23	23, 18, 0
24	24, 23, 22, 17, 0
25	25, 22, 0
26	26, 6, 2, 1, 0

© 1994-2020 All Rights Reserved.

Number of bits	Taps
27	27, 5, 2, 1, 0
28	28, 25, 0
29	29, 27, 0
30	30, 6, 4, 1, 0
31	31, 28, 0
32	32, 22, 2, 1, 0

The taps are numbered from left to right, starting with zero and going through N where N is the number of bits in the shift register.

© 1994-2020 All Rights Reserved.

Below is an example with 0 to 100% risetime, linear edge type, with a 32 bit MLS set.

© 1994-2020
All Rights Reserved.

When this runs in the transient, first the alternating pattern of bits defined as the starting state of the shift register is produced, and then the sequence begins.

© 1994-2020
All Rights Reserved.

Looking closer, the transition begins at exactly 1nSec.

Specify Bit File LSFR Mode

In this mode, an ASCII file with a sequence of 1 and 0 states is read. In this way, any arbitrary sequence can be used in the PRBS source. The states in the file are read from left to right,

© 1994-2020 All Rights Reserved.

and if the time of the simulation exceeds the number of bit states in the file, the pattern will repeat. An example of a bit file is shown below.

© 1994-2020 All Rights Reserved.

When the file with the bit pattern is available, set the LFSR mode to *Specify bit file*, and use the file browser to select the file. If you specify the filename manually, enter the complete path to the file for the file name. An example with the external bit file is shown below.

© 1994-2020
All Rights Reserved.

Note that the pattern in the file is reproduced, and repeats, as shown below.

Specify Seed and Taps LFSFR Mode

To use this feature, you must set the *Seed* and *Tap* information. The general configuration of a 6-bit Fibonacci shift register is as follows:

© 1994-2020
All Rights Reserved.

The boxes on the top are the shift register bits. The non-inverters in the middle can either be switched on or off, and are controlled by the taps vector. Taps zero and N (The taps at both ends of the shift register) are always on, which enables the circular connections. The shift register is clocked every period, and the bit shifts right. Depending on the setting of the feedback taps, different bits of the shift register can change the next bit that is loaded into the beginning (left) of the shift register. This logic is asynchronous. The adders at the bottom do modulo-2 addition, which is equivalent to an exclusive-or logic function.

Important

There are different conventions for numbering the taps. Please note that in Spectre, the taps are numbered from left to right, starting at zero. The bits in the shift register are numbered from left to right starting at one.

Seed

This is a list of bit numbers to set to the one state that are used to preload the shift register at time zero. An example is 1 3 5, which would load the first, third, and fifth bits with the one state, and the rest of the bits to the zero state. If the size of the shift register is smaller than the data, the higher bit numbers are simply discarded. The default is to set all the odd bits to the one state.

Taps

This is a list of the taps to set to one, and also defines the length of the shift register. In the above example, the numbers 1, 2, 3, 4, and 5 can be specified. Additionally, 6 must be set to define the length to be 6 bits. The zero tap and the tap with the highest value in the *Taps* field are always set to 1. By convention, the first number is the number of bits in the shift register, and the next series of numbers are in descending order and control the gain of the taps. In Spectre, the numbers can be specified in any order, and the largest number sets the size of the shift register. For example, 6 4 2 (separated by spaces) specifies a 6-bit shift register with G2 and G4 set to one. The rest of the gains are set to zero, or an open connection. Also note that 6 4 and 4 6 give the same sequence. The largest number in the string sets the size of the shift register, and the rest of the numbers define which taps are on and off. If the *Taps* field has a single number, this defines the size of the shift register, and the values in the *Seed* field will endlessly circle through the shift register.

Detailed Example of Setting the Taps

To manually specify the seed and taps, you must first set the LSFR Mode to Specify seed and taps. Next, set the size of the shift register, and define the taps. In this example, the size is set to six, and the fourth tap is enabled. The setting of the Taps property is 6 4 as shown

© 1994-2020 All Rights Reserved.

below. To set the bits in the shift register to the digital pattern 000011, set 5 6 in the *Seed* property. This will load the one state into the rightmost 2 bits of the 6 bit shift register.

© 1994-2020
All Rights Reserved.

The internal bits get set, as shown below. When the tap is on, it is a connection. When it is off, it is an open. When the adders that are not shown have a zero on the top connection, they take the bit from the right, add zero, which just brings that bit to the left. The other adders just become wired connections.

The taps vector enables G4, which brings the fourth bit from the left to the adder for that section. The adder sees a zero and a 1, so the output is 1. This 1 will be loaded in to the shift register at the next clock event.

At the next clock, the 1 from the output of the adder is loaded into the leftmost bit, and the contents shift right.

The output of the right bit is still 1, and the 4th bit from the left is still 0, so the output of the adder stays at 1.

© 1994-2020
All Rights Reserved.

At the next clock, the 1 is loaded in the leftmost bit, and the data shifts right.

Now, the adder sees zero on both inputs, so the output changes to zero. This is loaded in the next clock, and the data shifts right.

If you look at the sequence of the bits on the right side of the shift register so far, 1 1 0 0 has been generated. The sequence is from left to right. The first bit is a 1.

Continuing this exercise, it can be shown that the bits on the output should be 1 1 0 0 0 0 1 1 0 0 1 1 1 1 1 1 0 0 0 0 for the first 20 bits.

Now assume a bit period of 1nsec, The first bit, which is a 1, is generated from zero seconds to 1 nsec. The second bit, which is a 1, is generated from 1nsec to 2nsec. The third bit, which is a zero, is generated from 2nsec to 3nsec, The first transition from a 1 to a zero occurs at

© 1994-2020 All Rights Reserved.

2nsec. The bit sequence above would generate transitions at 2n, 6n, 8n, 10n, and 16n, which is shown in the result below.

Jitter Generation

Random Jitter

The PRBS source can generate random jitter, periodic jitter, or a combination of both types of jitter. For periodic jitter, up to three periodic jitter sources can be specified. First, random jitter will be considered. The property for setting random jitter is *RJ(rms)*.

RJ(rms)

© 1994-2020 All Rights Reserved.

This property defines one standard deviation for random jitter to be generated in the PRBS output waveform. Random jitter with a gaussian distribution will be generated on the transitions of the output waveform. This example has 20 psec jitter added, as shown below.

© 1994-2020
All Rights Reserved.

RJ(seed)

This is an optional property that can be used to generate different jitter sequences. Specify an integer in this field. The default value is 1.

Next, this was run for 20 microseconds using the transient, and plotted in the waveform tool. To see the jitter, first select the trace, and then select *Measurements - Eye Diagram*.

© 1994-2020
All Rights Reserved.

The eye diagram assistant starts. The data has a periodicity of 1 nsec, To put all the transitions in the center of the eye, a 0.5 nsec *Start Time* is set, along with the 1 nsec periodicity of the output signal. The timing variations blur the transitions in the center.

© 1994-2020
All Rights Reserved.

To see where the wave is centered, a zoom operation was performed, and a vertical marker has been placed at 0.5 nsec.

In order to get standard deviation calculation, we need to modify the shift register so that it provides a periodic output. This is done in the example below by setting the seed to 13 5, which sets a 1 state in the first, third, and fifth bit, and then taps is set to 6, which rotates the pattern through the shift register endlessly. In this way, a square wave is generated with a

© 1994-2020 All Rights Reserved.

periodicity of 2 nsec. This periodicity is needed in order for the standard deviation functions to work.

© 1994-2020
All Rights Reserved.

Now the transient is run for a large number of cycles. in the example below, it is run for 20usec, which gives 10,000 cycles of shift register output.

© 1994-2020
All Rights Reserved.

Select the trace, click the right mouse button and select *Send To Calculator*. This starts the calculator. Select the *abs_jitter* function. Set the threshold as appropriate for your circuit. 0.5 is appropriate for this example. Also set the period of the waveform in the *Tnom* field.

Click *OK*.

© 1994-2020 All Rights Reserved.

Now, plot the signal. The jitter plots, as shown below.

© 1994-2020
All Rights Reserved.

Now add the *stddev* function in the calculator. Send the result to a table.

© 1994-2020 All Rights Reserved.

The result is very close to the 20psec setting in the `prbs` source.

Periodic Jitter

Periodic jitter can be generated in the `prbs` source by setting the number of periodic jitter sources, choosing the waveshape for each jitter source, setting the frequency of each periodic jitter source, and by defining the delta-T (peak) for each jitter source in the

© 1994-2020 All Rights Reserved.

PJ(amplitude) field. Jitter waveforms can be sinusoidal, triangular, or square wave. Below is an example of a single periodic sinusoidal jitter source at 500KHz with 100psec peak jitter.

© 1994-2020
All Rights Reserved.

Run in the transient, and then send the waveform to the calculator. Apply the *abs_jitter* function as before. The jitter waveform is shown below.

© 1994-2020
All Rights Reserved.

The curve seems to indicate only the negative jitter. An eye diagram was plotted to check the actual timing of the edges.

Note that the jitter is actually centered at 1nsec, which is the timing of the noise-free waveform. The *abs_jitter* function takes the time of the first transition of the time-domain waveform as the center time of the jitter calculation. As a result, the center value can be skewed if the timing of the first edge is not noise free.

© 1994-2020 All Rights Reserved.

Below is an example of a triangle wave with 100psec peak jitter.

© 1994-2020
All Rights Reserved.

Again, with an eye diagram, the transitions are centered around 1nsec.

© 1994-2020
All Rights Reserved.

Two Periodic Jitter sources at the Same Time

Up to three periodic jitter sources can be set at the same time. Specify the number of periodic sources you want, and define the jitter parameters for each jitter source. An example with two jitter sources is shown below.

© 1994-2020 All Rights Reserved.

© 1994-2020
All Rights Reserved.

The resulting abs_jitter plot is shown below. Both jitter sources are in the output.

© 1994-2020
All Rights Reserved.

Random and Periodic Jitter at the Same Time

Both random and periodic jitter can be added at the same time. Below is an example of setting both in the `prbs` source.

© 1994-2020
All Rights Reserved.

The *abs_jitter* plot shows both random and periodic jitter

.

PRBS Mode External Triggering

When the `prbs` source is externally triggered, the schematic Pcell redraws for the source, and two control terminals are added to the left of the source. This is shown below.

When the source is externally triggered, periodic and random noise generators are turned off. The properties form specifies the voltage level for the trigger and the edge mode for the

© 1994-2020
All Rights Reserved.

trigger. In the example below, the trigger threshold is 0.5 Volts and the trigger is on the rising edge.

© 1994-2020
All Rights Reserved.

Note that the jitter terms are greyed out, indicating that the jitter is not available in the external trigger mode.

When the threshold is reached on the input waveform, the output begins the transition to the opposite state. The timestep is actively controlled in the region of the trigger threshold to make the signal trigger very accurately. In the waveform window below, the blue trace is the input, and the trace symbols are on and mark the timesteps actually used in the simulation. As the threshold is reached, the timestep is actively managed so that the crossing is accurately detected.

The jitter is also very small, as seen below.

The calculator expression is shown on the left, and this is the same expression used before in this section to calculate the jitter.

© 1994-2020
All Rights Reserved.

PRBS Source with External Triggering Using a Pulse or PWL Signal

The external triggering capability of the `prbs` source depends on being able to calculate the slope of the input signal in order to accurately trigger at the specified voltage. The pulse and pwl sources guarantee timepoints at the corners of the waveforms, but they do not guarantee timepoints during the transition, especially if the transition is short. In that case, the prbs will trigger on the timepoint on the end of the transition as shown below.

The threshold for the trigger is 0.5 volts, and the input signal is the red trace. The actual timepoints of the input (red trace) are turned on, and the output starts its transition when the input is at 1 volt instead of at 0.5 Volts. This adds a slight delay in the prbs output, which in many cases is not significant. If you need to trigger on the actual threshold, then add a resistor

© 1994-2020 All Rights Reserved.

and capacitor across the source as shown below. Set the R*C product to be the risetime of the input signal.

Adding these components adds numerical integration, and the error of this integration is controlled by Spectre by adding appropriate timepoints. This establishes a slope of the input signal so the trigger event can happen at the desired threshold of 0.5 volts in this example as

© 1994-2020 All Rights Reserved.

shown below. The green trace below is the junction of the resistor and capacitor that was added to the circuit. Again, the timepoints are turned on for the input trace.

Keep the R*C product equal to the risetime of the pulse source or slightly less so the timestep is not excessively reduced for a long period of time.

© 1994-2020
All Rights Reserved.

Oscillator Macro Parameters

The *oscmacro* source can be used to generate oscillator waveform and noise in pss-pnoise and hb-hbnoise analysis. The *oscmacro* source does not generate noise in transient analysis.

File Name

Specifies the file name containing the harmonics and noise data results from hb and hbnoise analyses. For more information, see Creating an Oscillator Macro Source.

© 1994-2020
All Rights Reserved.

Freq scale

Specifies the frequency scaling factor. This is a multiplier for the frequency of the source. The default value is 1.0 which uses the frequency in the specified file.

Amplitude scale

Specifies the amplitude scaling factor. This is a multiplier for the amplitude of the output. The default value is 1.0 which uses the amplitude in the specified file.

Noise Parameters

Important

The check box is a display feature in the form only. If there are properties that are set and the display check box is not selected, although you will not see the parameters in the form, these parameters will still be netlisted, and will still have an effect on the simulation.

The noise parameters include *Noise temperature*, *Noise Entry Method*, *Noise file name, Noise type, Interpolation method*, and *Noise/Frequency points*.

The *Noise type* selection sets one of two basic modes. When *Noise type* is set to *Noise Voltage*, pure noise voltage is injected for the noise analysis. In this mode, you specify frequency-volts-squared per Hertz pairs either in a file, or manually entered into the form. The second mode is to inject pure phase noise. In this mode, you specify frequency—ssb phase noise in dBc/Hz pairs. Phase noise is only available when the *Source type* property is set to *sine*.

Voltage noise is observed in transient noise, but phase noise is not.

Important

This mode can be tricky. Remember that for all noise analyses, the noise specified will be extrapolated outside the frequency range that is specified. When pnoise or hbnoise is used, this extrapolation can cause noise at harmonics of the large-signal frequency, which may result in more noise than expected. Make sure that you specify a noise frequency below the lowest frequency with zero as the noise setting, and specify an additional noise point slightly above the last point with the noise set to zero.

© 1994-2020 All Rights Reserved.

The first example shown below has voltage noise injected. The default linear interpolation should be used when the noise data has linear spacing. If the noise data has log spacing, log-log interpolation will cause the noise analysis to be more accurate.

When phase noise is selected, log-log interpolation is always used. The frequency terms for the phase noise data should be the offset frequency from the carrier. The noise term should

© 1994-2020
All Rights Reserved.

be single-sideband phase noise (ssbphasenoise) in dBc/Hz. The phase noise will be injected above and below the frequency for the first sinusoid the source produces.

Important

This mode can be tricky. Remember that for all noise analyses, the noise specified will be extrapolated outside the frequency range that is specified. When pnoise or hbnoise is used, this extrapolation can cause noise at harmonics of the large-signal frequency, which may result in more noise than expected. Make sure that you specify a noise frequency below the lowest frequency with minus 400 as the noise setting, and specify an additional noise point slightly above the last point with the noise set to minus 400 as well.

© 1994-2020 All Rights Reserved.

An example of phase noise is shown below

.

© 1994-2020
All Rights Reserved.

Noise temperature

The *Noise temperature* of the *port* resistor in degrees Celsius. If this is not specified, the *Noise temperature* is assumed to be the temperature that is set by the temperature option. When you compute the noise figure of a circuit driven at its input by a *port*, set the *Noise temperature* of the *port* to 16.85C (290K). This setting matches the standard IEEE definition of noise figure. In addition, disable all other sources of noise in the *port*, such as the Spectre parameters *Noise file name* and eliminate any entries you have in the fields that appear when you select *Noise/Frequency points*. If you want a noiseless *port*, set the *Noise temperature* to absolute zero or below, and do not specify a noise file or Noise/Frequency points. Absolute zero is -273C. The default noise temperature is 290C.

Noise Entry Method

You can select one of the two ways to enter noise data, either by specifying a *File* name or entering a series of *Noise/Frequency points*.

Important

Entering noise points define AM noise only. There is no mechanism in the port for including PM noise from a noise file.

Noise file name

If *Noise Entry Method* = *File* is selected, you enter the name of the file containing the noise data in the form of frequency-noise pairs. In your file, list the frequency-noise pairs as one pair per line with a space or tab between the frequency and noise values. The first entry is the frequency, and the second value is volts-squared per Hertz.

Important

Frequencies outside the range specified in the noise file are interpolated, and not zero. If you want zero, make sure that you specify zero over the range you want. Remember that in pnoise, a wide range of noise frequencies are typically evaluated, and this interpolation can cause more noise than expected in the analysis. If you want to see what the source is actually generating, make a circuit with a port and a terminating resistor, and run linear noise over a very wide frequency range. Then plot the noise at the output of the port.

© 1994-2020 All Rights Reserved.

Important

If you want just the noise specified in the file and nothing more, set the noise temperature of the port to -273. The default is to generate the noise from the file and add to it the noise from the resistor in the port. Setting the noise temperature to -273 makes the noise fro the resistor be zero.

Figure 8-13 Display noise parameters: Noise Entry Method=File

Important

The check box is a display feature in the form only. If there are properties that are set and the display check box is not selected, although you will not see the parameters in the form, these parameters will still be netlisted, and will still have an effect on the simulation.

Important

If you want just the noise specified in the noise file and nothing more, set the noise temperature of the port to -273. The default is to generate the noise from the file and add to it the noise from the resistor in the port. Setting the noise temperature to -273 makes the noise from the resistor be zero.

Num. of noise/freq pairs

If the *Noise Entry Method* is set to *Noise/Frequency points*, you specify the *Number of noise/freq pairs*. The form expands to let you type in the designated *Freq* and *Noise* values. The noise values must be in volts-squared per Hz, and frequency in Hz. The maximum number of points in the form is 10.

The example below has the *Number of noise/freq pairs* set to 3.

© 1994-2020 All Rights Reserved.

Figure 8-14 Display noise parameters: Noise Entry Method=Noise/Frequency Points

Important

If you want just the noise specified in the *Num. of noise/freq pairs* properties and nothing more, set the noise temperature of the port to -273. The default is to generate the noise from the noise properties and add to it the noise from the resistor in the port. Setting the noise temperature to -273 makes the noise from the resistor be zero.

Small-Signal Parameters

Display small signal params

Important

The check box is a display feature in the form only. If there are properties that are set and the display check box is not selected, although you will not see the parameters in the form, these parameters will still be netlisted, and will still have an effect on the simulation.

If selected, the *Edit Object Properties/Add Instance* form expands to show the small-signal properties *PAC Magnitude*, *PAC Magnitude (dBm)*, *PAC phase*, *AC Magnitude*, *AC phase*, and *XF Magnitude*.

© 1994-2020 All Rights Reserved.

Figure 8-15 Display small signal params

When you specify the voltage on a *port*, you are specifying the voltage when the port is properly terminated, and not the voltage on the internal voltage source. Thus, the voltage on the internal source is set to twice the value specified on the *port*. The same is true for the values for the transient, AC, and PAC signals. However, the amplitude of the sine wave in the PAC and transient analysis can alternatively be specified as the power in dBm delivered by the `port` when terminated with the reference resistance. The *PAC magnitude* property is used by pac, hbac, and qpac. The *XF Magnitude* property is used by xf, pxf, hbxf, and qpxf.

PAC Magnitude

The peak periodic AC analysis magnitude. Setting this value to unity is a convenient way of computing the transfer function from this source to the output.

You can select either *PAC magnitude* or *PAC magnitude (dBm)*, but not both. If *PAC magnitude* has a value, the *PAC magnitude (dBm)* field is grayed out. The PAC magnitude property is used by pac, hbac, and qpac.

PAC Magnitude (dBm)

The periodic AC analysis magnitude in dBm (alternative to *PAC magnitude*). You can select either *PAC magnitude* or *PAC magnitude (dBm)*, but not both. If *PAC magnitude (dBm)* has a value, the *PAC magnitude* field is grayed out. The *PAC magnitude(dBm)* property is used by pac, hbac, and qpac.

PAC phase

The periodic AC analysis phase. The value must be a real number. The default value is 0 and the units is in degrees.

© 1994-2020 All Rights Reserved.

Typically, only one source in the circuit has a *PAC magnitude* set to a value other than zero, and many times it has a *PAC magnitude*=1 and *PAC phase*=0. However, there are situations where more than one source has a nonzero *PAC magnitude*. For example, applying a differential small-signal input could be done with two sources with the *PAC magnitude*s set to 0.5 and the *PAC phase*s set to 0 and 180. The *PAC Phase* property is used by pac, hbac, and qpac.

You do not specify the PAC frequency in the port *Edit Object Properties* form. Instead, you set the PAC frequency in the PAC *Choosing Analyses form*. For example, when making an IP3 measurement, you set the PAC frequency to a variable value in the *Choosing Analyses* form. Then, you enter the same variable in the *PAC Amplitude* (or *PAC Amplitude dBm*) field of the *port Edit Object Properties* form.

AC Magnitude

The peak small-signal voltage for the ac analysis in volts.

AC phase

The ac small-signal phase in degrees.

Typically, only one source in the circuit has *AC Magnitude* set to a value other than zero, and many times it has an *AC magnitude*=1 and *AC phase*=0. However, there are situations where more than one source has a non-zero *AC magnitude*. For example, you can apply a differential small-signal input with two sources with the *AC magnitude*s set to 0.5 and the *AC phase*s set to 0 and 180.

XF Magnitude

This property is seldom used. It specifies the transfer function analysis magnitude. This is used by xf, pxf, hbxf, and qpxf.

Temperature Effect Parameters

Temperature effect parameters include *Linear temp. coefficient*, *Quadratic temp. coefficient*, and *Nominal temperature*.

© 1994-2020 All Rights Reserved.

Display temperature params

Important

The check box is a display feature in the form only. If there are properties that are set and the display check box is not selected, although you will not see the parameters in the form, these parameters will still be netlisted, and will still have an effect on the simulation

If the check box is selected, the *Edit Object Properties/Add Instance* form expands and the following three parameters appear in the form: *Linear temp. coefficient, Quadratic temp. coeff.,* and *Nominal temperature*.

Figure 8-16 Display temperature params

Linear temp. coefficient

This property sets the first order (linear) temperature coefficient of all the voltages specified for any waveform in the port. 0.01 entered in the *Linear temp. coefficient* field is interpreted as a one percent change per degree Celsius. If the number is positive, the voltages increase with temperature. If the number is negative, the voltages decrease with temperature. If the temperature moves far enough, the values can become negative.

Quadratic temp. coeff.

This property sets the second order (quadratic) temperature coefficient of all the *voltages* the source generates. 0.01 causes a change in value of one percent per degree squared.

Nominal temperature

The *Nominal temperature* (tnom) for the linear and quadratic temperature coefficients.

© 1994-2020 All Rights Reserved.

Simulating Tabulated S-Parameters Using the Nport Component

Nport from analogLib provides a generic way to incorporate an S-Parameter file into a circuit simulation. The nport component allows you to set the number of ports, and when this is done, the schematic symbol re-draws with the number of ports shown. There is also n1port, n2port, n3port, and n4port in analogLib, which have a fixed number of ports. These components are provided for legacy purposes only, and should not be used in current simulations.

For the frequency-domain based analyses, such as ac, noise, xf, stb, pz, harmonic balance, hbac, hbxf, and hbnoise, the S-Parameter file is used directly in the frequency domain. For the DC and transient based simulations dc, tran, and shooting, a time-domain model must be created from the frequency-domain description in the S-Parameter file. This is a non-trivial process.

Note: Currently, the improvements to the nport component are restricted to transient analysis only.

In general, there are two different ways to create a time-domain model from an S-Parameter file. One way is to do a rational fit, which attempts to match the frequency domain description using a combination of poles, zeros, and gain blocks. The other way uses a convolution-based approach. This approach calculates the impulse response, and then creates a time-domain model based on the impulse response.

Regardless of the method, there are fundamental requirements for the S-Parameter file. These are:

- The file must contain a frequency point that is low enough so that when the DC point is extrapolated, the DC point correctly defines your circuit behavior.

  This is a fundamental requirement. To run a transient, first you need to run the DC analysis to capture the time-zero timepoint. If the first frequency in the file is not zero frequency, then a zero-frequency point must be extrapolated. If the first frequency in the S-Parameter file is not low enough to accurately represent the real DC characteristics of the system, the DC extrapolation will be incorrect.

- The phase of the DC point must be zero or 180 degrees.

- The frequency-domain data must be dense enough to accurately describe the transfer function.

  Smooth transfer functions without local complex behavior might need relatively few datapoints. Systems with complicated behavior need to have enough datapoints to

© 1994-2020 All Rights Reserved.

capture the behavior well enough so that when the data is interpolated between the given points, and the interpolation error is small.

For bandpass filters, make sure that data from near zero Hz through at least the frequency of the highest harmonic in the simulation is present in the S-Parameter file.

- The frequency data must exist to a high enough frequency so that harmonics of the system are accurately simulated.

  If your S-Parameter file has data only through the second harmonic of the input signal and you are calculating six harmonics in a large-signal analysis, just note that the S-Parameter data above the second harmonic is extrapolated. The simulation is not based on real data, and as a result, it is different from the real system. If you want your simulation to be accurate for the harmonics you are simulating, make sure that the S-Parameters themselves go high enough in frequency.

- As a strong suggestion, regardless of the source of the data, plot the data from your S-Parameter file before you simulate. The plot for all the S-Parameters should plot in a clockwise arc with no sudden jumps. These discontinuities cause large inaccuracies in the transient model. Spectre now checks for discontinuities in the S-Parameter data.

To plot an S-Parameter file directly in the waveform tool, the file format must be in Touchstone1 format. Here are the steps required to plot an S-Parameter form an S-Parameter file.

1. From the ADE Explorer window, select *Tools-Waveform*.

© 1994-2020 All Rights Reserved.

2. In the waveform tool, select *File-Open Results*, and browse to the location of the S-Parameter file.

© 1994-2020
All Rights Reserved.

3. When the file is selected, the waveform tool has the path to the results at the top-left and the S-Parameter filename at the bottom. If you started from ADE Explorer, you may also have the path to the psf file on the upper left.

© 1994-2020
All Rights Reserved.

4. Double-click the filename. The S-Parameters in the file are displayed.

© 1994-2020
All Rights Reserved.

5. On the upper left, select *Impedance* or *Admittance*. This tells the waveform tool to plot on a Smith Chart.

© 1994-2020
All Rights Reserved.

6. Click one of the S-Parameters. Click the right mouse button and select *Move to Plot Signal*. The signal plots.

Important

S11 above has visible piecewise-linear behavior at low frequencies. This S-Parameter file will produce incorrect time-domain results at the frequencies between the inflection points in the S-Parameter plot.

Potential Problems with S-Parameter Files

Not all frequency domain descriptions are causal. This is a fundamental requirement for the time-domain model to be accurate. When Spectre reads in an S-Parameter description and causality is being checked, the transfer function from the highest frequency in the S-Parameter file through three times the highest frequency in the S-Parameter file is adjusted to enforce causality. Frequencies above three times the frequency in the S-Parameter file should be avoided.

© 1994-2020 All Rights Reserved.

Convolution-Based Method

The convolution-based method is strongly suggested for most simulations. This method is used when the interpolation method is set to linear or spline. The default interpolation method is linear, and the convolution-based method is the default.

The convolution-based method samples the frequency-domain data, and then calculates the impulse response for that frequency description. Then a Fourier transform is used to create the DC and time-domain model.

Considerable development has been made in MMSIM12.1 so that the only thing that should be required is to set the S-parameter file name, the interpolation method, and the number of ports. The default settings are strongly recommended for all other parameters. By default, the impulse responses are cached. This allows the reuse of previously computed impulse response files, thereby saving a lot of time in simulations after the first time the simulation is run. If multiple instances in the schematic use the same S-Parameter file, the impulse response is calculated only once. The impulse response may need to be recalculated if you specifically set some of the properties that are available in the nport. Except for the *Passivity* parameter and the *Accuracy* parameter, changing any other property is highly discouraged unless you are an expert in simulating S-Parameters. The old Impulse response will not be erased. If you switch back to the previous settings in the nport, those files will still be available, and will still be reused. The location of the files depends on the setting of the *nportfiledir* option which is documented later.

S-Parameter files with a large number of ports require lengthy calculations. In many cases, although there are a large number of ports in the S-Parameter file, only a small number of ports are actually connected in the circuit. This is common for package models. Starting with the MMSIM 12.1.1 release, the number of calculations can be reduced by eliminating the unused ports from the circuit, thus, improving the simulation time. This is done at the beginning of the simulation, and affects all the analyses that are specified to be run. If there is a component in the circuit connected to a port in the circuit, it is kept. If there is no connection, this port is removed. If you desire to monitor the output of a port that is not connected, add a large resistor to ground on this port so you can see what appears on this connection point. By default, if there are 20 ports or more, and 80% or less are actually connected to other components, the nport will be compressed. There are also global options in the *Simulation* - *Options* - *Analog form* (*Components* tab) to set compression explicitly.

Rational Fit Methods

There are two rational fit methods. In general, these methods fit the data using poles and zeros to match the S-Parameter data. Of the two methods, bbspice is preferred. Bbspice is used to do the rational fit. The product bbspice has many desirable features such as non-

© 1994-2020 All Rights Reserved.

causal data identification that are not implemented in the nport. For this method, set the *Interpolation method* to *bbspice*.

© 1994-2020 All Rights Reserved.

Using the nport Component

The nport symbol is a schematic Pcell, which redraws when you change the number of ports. The symbol has two different modes. At the top of the properties list, is a selection called *Common reference*, which is a check box. If the item is not checked, plus and minus pins will be displayed for all the individual ports. If the check box is selected, then the symbol redraws with a single common ground reference pin at the bottom of the symbol. This eliminates the need to add ground connections to each port of the symbol. The check box is highlighted in the properties list below.

Figure 8-17 nport Component Edit Object Properties Form

© 1994-2020
All Rights Reserved.

The symbol without the selection is shown on the left in the figure below, and the symbol with the selection is shown on the right.

Use the *nport* component to read in S-parameter data. Follow these steps to prepare the component for use in SpectreRF simulations.

© 1994-2020
All Rights Reserved.

Figure 8-18 nport Component Edit Object Properties Form

1. Select the *nport* component from *analogLib*.
2. Place the *nport* component in the schematic window.
3. Set whether you want a single ground terminal on the nport symbol or not.
4. Set the number of ports that are in the S-Parameter file.
5. In the Schematic window, highlight the new component and then choose *Edit – Properties – Objects, or use the bindkey* you have set up for setting properties.

   The *Edit Object Properties* form for the *nport* component is displayed.

© 1994-2020 All Rights Reserved.

6. Click the file browser button located to the right of *Browse and select the s-data file* in the properties form. From the browser, select the S-Parameter file.

Alternatively, type in the S-Parameter data filename. If you specify just the filename in the port component (not the complete filename starting from slash), then in ADE Explorer, select *Setup - Simulation Files*, and enter the path to the directory that contains the S-Parameter file in the *Include Paths* section. This is necessary so that the file can be found.

© 1994-2020
All Rights Reserved.

Note: You can also specify an S-parameter file as a design variable. To do this, select the *S-parameter file as Design Var?* check box and enter the name of the variable in the *S-parameter data file* field. However, this action does not add the file in ADE Explorer automatically even when you select the *Variables - Copy from CellView* menu option. You need to manually add a design variable in ADE Explorer and enter the file name `"/<path>/<file_name>.s2p"` surrounded by quotes in the *Value* field.

7. Although the properties form is labeled S-Parameter data file, it can also be a Y-Parameter or Z-Parameter file. Spectre recognizes all three formats. It is labeled this way because S-Parameter files are more common than Y or Z-Parameter files.

8. Set the *Interpolation method* cyclic field.

   The *linear* and *spline* values are usually preferable to *rational.* If you want a rational fit, *bbspice* is preferred over rational. In general, the recommended method is *linear.*

   Linear and spline control the sampling of the S-Parameter data for the convolution-based method. In both methods, the S-parameter data is sampled using a linear frequency spacing from zero to three times the highest frequency in the S-Parameter data file in order to calculate the impulse response of the transfer function.

Important

Note that when the S-Parameter file is for a physically large structure like a package model, because of the physical size of the structure, the inductance drops starting in the range of 100KHz to 1MHz. To correctly model this behavior, you require data in the S-parameter file that accurately describes this behavior, and it requires a huge number of frequency sample points in order to get a correct time-domain model.

When *linear* is selected as the interpolation method, linear interpolation is used to get a datapoint needed in the sample that is not directly in the S-Parameter file. Spline uses a cubic spline algorithm. Cubic spline can occasionally introduce errors when there are rapid changes in the transfer functions defined in the S-Parameter file near the sample point.

When the data in the S-Parameter file is dense, linear and spline produce comparable results. When the data in the S-Parameter file is sparse, non-physical results can be introduced because of the wiggles that can be introduced by a polynomial fit. Linear is usually a better choice when the data is sparse.

When you select *rational*, four new fields are displayed. For information about these fields, see Controlling Rational Fit Accuracy on page 1889.

9. Specify whether you want the passivity check to run. Setting *Passivity* to *no* causes no passivity check to be run. Setting *Passivity* to *check*, which is the default, will run a

© 1994-2020 All Rights Reserved.

passivity check, and the result of the check will be sent to the Spectre output window. The S-Parameter file will be used as specified.

If *Passivity* is set to *enforce*, the passivity check will be run, and the S-Parameter file will be changed as necessary to ensure the passivity of the result. Note that this will change the S-Parameters as specified in the file if a non-passive entry is detected. The *no*, *check*, and *enforce* options are applicable only when the interpolation method is set to *linear*, *spline*, or *rational*.

If *Interpolation Method* is set to `bbspice`, passivity may be set to *fit_weak_enforce* or *fit_enforce*. The default value is *fit_enforce* and with this setting the simulator always attempts to create a passive simulation model and favors passivity over accuracy. When *Passivity* is set to *fit_weak_enforce*, the simulator does not create a passive model if the passivity enforcement phase results in a significant accuracy loss.

10. If exceptional accuracy is desired, select *Conservative* for the Accuracy default. This will cause more frequency-domain sample points, and produce a more accurate impulse response at the cost of runtime. *Default* accuracy is set tight enough that *Conservative* should only be needed in rare instances.

© 1994-2020 All Rights Reserved.

When the DC or transient-based simulation is run, Spectre provides information about the S-Parameter file and the impulse response file in the output window.

```
/lan/csv/rfrd_v2/user/rdavis/rdavis/demos/ic61/docs_database/simulation/nport_Test/sp
File  Help                                                                    cādence

Warning from spectre during initial setup.
    WARNING (CMI-2831): In file `graphs.s2p', S1_1 may be discontinuous at 608.2 MHz.
    WARNING (CMI-2831): In file `graphs.s2p', S1_2 may be discontinuous at 608.2 MHz.
    WARNING (CMI-2831): In file `graphs.s2p', S2_2 may be discontinuous at 608.2 MHz.
Notice from spectre during initial setup.
    S parameters are passive in file `graphs.s2p'.

Circuit inventory:
              nodes 2
              nport 1
               port 2

Time for parsing: CPU = 0 s, elapsed = 9.88889 ms.
Time accumulated: CPU = 109.982 ms, elapsed = 339.951 ms.
Peak resident memory used = 35.6 Mbytes.

Warning from spectre.
    WARNING (SPECTRE-16707): Only tran supports psfxl format, result of other analyses

***************************************************
Transient Analysis `tran': time = (0 s -> 5 ns)
***************************************************

Warning from spectre during DC solution estimation, during IC analysis, during transien
    WARNING (CMI-2134): Risky extrapolation to DC of data given in S-parameter file `gr

Important parameter values:
    start = 0 s
    outputstart = 0 s
    stop = 5 ns
    step = 5 ps
    maxstep = 50 ps
    ic = all
    useprevic = no
    skipdc = no
    reltol = 100e-06
    abstol(V) = 1 uV
    abstol(I) = 1 pA
    temp = 27 C
    tnom = 27 C
    tempeffects = all
    errpreset = conservative
    method = gear2only
    lteratio = 10
    relref = alllocal
    cmin = 0 F
    gmin = 1 pS

Notice from spectre during transient analysis `tran'.
    Nport NPORT0: Reuse impulse responses from the file `9dbf58e44917aadedff09cb933bba7

    tran: time = 126.6 ps     (2.53 %), step = 4.21 ps      (84.2 m%)
4  HelpAction
```

Nport Compression

In cases where package models are used, many ports may be specified in the S-Parameter file, but only a small number of ports are actually connected to the circuit for simulation. Although the nport caches the data from the S-Parameter file, with a large number of ports, the extraction time can be an issue. To speed these extractions, the S-Parameter file can be

© 1994-2020 All Rights Reserved.

compressed, which removes the unused ports from the extraction and from the simulation result, but causes the extraction to be considerably faster. The extraction time is reduced to the square of the connected port fraction. If 10% of the ports are connected, the extraction time goes to 0.01*old extraction time.

Unused ports are ports that are not connected to any circuit component, ports that are connected to ground, or ports that are shorted together.

By default, if there are 10 or more ports in the S-Parameter file, and less then 80% of the ports are connected, nport compression runs. First, the connections to ground and shorted ports are applied, then an s-parameter analysis runs to capture the performance of the connected ports. That file is written to the netlist directory by default, and has a global option to control the location. To find the netlist directory, first determine the location of the simulation directory by selecting *Setup - Simulator/Directory/Host* menus in ADE Explorer, and read the location of the *Project* Directory. Navigate to this directory, then type *cd <circuit_Name>/spectre/<schematic_or_config>/netlist*. The compressed filename is the original S-Parameter filename with an extension. This is the file that is actually used for the simulation.

Note that this occurs before any of the analyses are run, so all the simulation results from all the analyses will have data for only the ports that have circuit connections. If you want to monitor the result on a port that you do not want to connect to the circuit, add a large resistor to ground that is connected to that port so that the data for that port is saved.

Controlling the Directory for the Compressed S-Parameter File

This directory can be set using the global options form by selecting *Simulation - Options - Analog*, and then selecting the *Component* tab. Specify a full pathname to the desired directory location in the *nportcompressfiledir* field.

Compression can also be controlled manually by selecting *yes* or *no* for the *nportcompress* option.

© 1994-2020 All Rights Reserved.

These options are shown below.

nportunusedportgmin

This option adds resistors to nport pins that are not connected to circuit components. Setting a small value (on the order of 1e-9 to 1e-3) adds a resistor with conductance set to this value from each unconnected pin to ground. This introduces a small error, but adds a small amount of damping to the nport response. This is sometimes useful for obtaining a passive response.

© 1994-2020 All Rights Reserved.

nportunusedportrmin

This option adds resistors in series with nport pins that are shorted together. Setting a small value (on the order of 1e-3 to 1 ohm) adds a resistor with resistance set to this value in series with the connection between pins. This introduces a small error, but adds a small amount of damping to the nport response. This is sometimes useful for obtaining a passive response.

© 1994-2020 All Rights Reserved.

Advanced transient convolution parameters

These properties are provided for S-Parameter simulation experts only.

© 1994-2020
All Rights Reserved.

Max sampling points

Max sampling points defines the maximum number of frequency points to be sampled in the adaptive algorithm. The default is 131072. In every case, encountered so far, the actual number of samples taken by the adaptive algorithm is much smaller than the default. In extremely unusual cases, it can be raised to 262144.

Max frequency of interest

The property *Max frequency of interest* should not be changed unless you are an expert in the simulation of S-Parameter files.

Max frequency of interest (fmax) controls the highest frequency for the frequency domain sampling of the S-Parameter file. Leave this property at the default. The default is three times the highest frequency in the S-Parameter file.

Impulse response truncation

Impulse response truncation is used to deliberately cut off the tail of the impulse response which might theoretically continue to infinite time. Leave this property at the default of 1e-4, which corresponds to a gain of -80 dB.

Causality Correction

The *Causality Correction* drop-down list contains three choices: *no*, *fmax*, or *auto*. *fmax* is the default and is highly recommended. Causality enforcement is required in order to have reasonable results from an nport in either the DC or transient-based analyses. Causality correction is performed by setting the transfer function between the highest frequency in the S-Parameter file and three times this frequency so that the data becomes causal. The data within the frequencies specified in the S-Parameter file is unchanged. Setting causality to `fmax` or `auto` overrides the setting of the high frequency extrapolation property.

- *no* does not add a causality check.

Important

Setting the causality check to *no* is incredibly risky unless you are absolutely sure that the S-Parameter file is causal as described.

- *fmax* retains the data in the frequency range of the S-parameter file, and then adds a transfer function above the frequency range in the S-parameter file to force the system to be causal. This transfer function extends to the setting of *Max frequency of interest*, which defaults to three times the highest frequency in the S-Parameter file. If you suspect

© 1994-2020 All Rights Reserved.

that the maximum frequency of interest needs to be changed, use causality *Auto* instead, if you are not an expert.

- *auto* applies the causality correction in a similar manner to choosing *fmax*. *auto* can also vary the maximum frequency of interest if it needs to get a causal time-domain model.

DC extrapolation

The *DC extrapolation* can be set to *constant* or *unwrap*. *constant* projects the first point down to zero frequency at exactly the same level. *unwrap* does an estimation based on the first few frequency points in the S-Parameter file. The default is *constant*.

Noise Parameters

These properties should never be needed.

Thermal noise

Thermal noise defaults to *yes*. If you want to disable noise production, set to *no*.

Thermal noise model

Thermal noise model defaults to external, which reads the noise parameters in the S-parameter file if it is available, and if not, it uses an internal noise model. Internal forces the internal noise model.

Noise correlation

The default is complex. The noise parameters in the S-parameter file might have real and imaginary parts. By default, if these parameters are complex, Spectre observes them. In some cases, passive components may have complex noise parameters in the noise file, which is theoretically not possible. In this case, set *Noise correlation* to *real*.

© 1994-2020 All Rights Reserved.

Rarely Used Parameters

These properties should never be needed.

© 1994-2020
All Rights Reserved.

S-Parameter data format

S-Parameter data format is not needed because Spectre can directly parse an S, Y, or Z-parameter file in Spectre, Touchstone1, Touchstone2, CITI file, RFM, or Sigrity BNP format file.

Note that the Touchstone1 file format now provides support for multiple impedances at each port. This is done in the header of the file. When R is listed, then multiple values can be entered up to the maximum number of ports in the file. If a number is omitted, then that number takes on the value of the first R entry.

For example, `#GHz RI R 50 1 1 50` means the frequency numbers are in GHz. The data is in real-imaginary format. The resistance is 50 Ohms on port 1 and 4, and 1 Ohm on ports 2 and 3.

`#GHz RI R 50 10 20 30` means 50 ohms on port 1, 10 ohms on port 2, 20 ohms on port 3, and 30 ohms on port 4.

`#GHz RI R 50 10` means 50 ohms on ports 1, 3and 4, and 10 ohms on port 2.

Causal s-param output file

Enter a filename beginning with slash (/) in this field. Specifying a filename causes the S-Parameter data after causality correction to be placed in this file. This file will be created the first time a tstab or transient simulation runs after the file is specified. This file can then be plotted directly in the waveform display tool. Most of the time, the data within the S-Parameter file frequencies is not disturbed, and the data between the highest frequency and three times the highest frequency is the data that is added to make the time-domain model causal. Below

© 1994-2020 All Rights Reserved.

is a comparison of the raw data and the causality-corrected data. The red trace is the original data.

© 1994-2020
All Rights Reserved.

This has the data plotted on a linear X-Axis.

This has the data plotted on the Smith Chart. The original data is in red, and the causality-corrected data is in blue.

© 1994-2020
All Rights Reserved.

Here is the original data in the Smith Chart.

Passivity tolerance

This property is only used when the *Passivity* property is set to *check* or *enforce*. Passivity tolerance does not need to be set. The default is 1e-6. This defines how close to unity gain should be modified by the passivity check and enforcement. Passivity will be enforced and/or reported when the gain is (`1 - Passivity tolerance`) or greater.

Data truncation threshold

Leave this property at the default of 1e-3, which corresponds to -60dB gain. When the cross coupling terms become smaller than the data truncation threshold, they are ignored. Cross coupling is the coupling from one port to another port.

Frequency sampling interval

Frequency sampling interval sets the delta frequency for the sampling from zero to the maximum frequency of interest. Leave this property at the default value. With adaptive

© 1994-2020 All Rights Reserved.

sampling, this should never be necessary. If used, this delta should be a power of two divisor of the maximum frequency of interest.

Multiplier

Multiplier specifies how many nport devices to put in parallel. This is rarely used.

Scale Factor

This scales the frequency of the S-Parameter file. For example, many S-Parameter files have the frequency in GHz. In this case, set the *Scale factor* to 1e9.

High freq extrapolation

This property should not be used. This property is ignored when causality correction is applied. The *High freq extrapolation* field can be set to *constant* or *linear*. If the causality check is not run, *constant* maintains the same amplitude and phase as the last point in the S-parameter file to infinite frequency. *linear* keeps the amplitude constant at the last frequency point, but the phase increases linearly with frequency.

Flag for matrix form input

Flag for matrix form of input should not be needed to be set. In the past, each time the simulation ran, the impulse response was calculated for every port of every instance of the nport every time the simulation was started. In some cases, especially with a large number of ports, this could take considerable time. This flag was provided so the step of calculating the impulse response could be skipped. Since the impulse response is cached and available for re-use at any time, this property should never be needed.

Additional parameter list

This field is typically used by beta customers to unlock new features. Note that when this feature is used, a warning message will be issued. This warning can be ignored.

Controlling Rational Fit Accuracy

Rational fitting is the process of adding multiple poles, zeros, and gain blocks to match the frequency response that is defined in the S-Parameter file. Note that the S-Parameter file may have distributed effects. Approximating these distributed effects with poles and zeros can

© 1994-2020 All Rights Reserved.

require a system with a very high order. Matching an arbitrary frequency-domain transfer function with poles and zeros is a difficult and complicated process.

Presently, the best rational fit option is bbspice. No controls are provided for this interpolation method.

At the current time, it is usually better to use the convolution-based algorithms when the interpolation method is set to *linear* or *spline*, or use the *bbspice* rational fitting method.

If you do use the rational fit, always test the rational model that is produced by running sp analysis using a circuit composed of an input port, an output port, and an nport with the

© 1994-2020
All Rights Reserved.

interpolation method set to *rational*. Run the sp analysis, and plot all the S-Parameters. Look for differences between the original S-Parameter file and the S-Parameters with the rational fit.

Also, the algorithm is known to take a very long time to run for S-Parameter files with a large number of ports.

The *nport* component has three parameters to control the accuracy of the rational interpolation process:

- *relerr* (The *Relative error* field)
- *abserr* (The *Absolute error* field)
- *ratorder* (The *Rational order* field)

You can use these parameters to trade off accuracy against model size and simulation time. In general, the more stringent the accuracy requirement, the higher the model order. Higher-order models require longer extraction and simulation time. SpectreRF can automatically generate a model that meets a specified accuracy requirement. You can also specify the model order directly to SpectreRF.

Using the relerr and abserr Parameters

The *relerr* parameter is used to control the accuracy of the rational fit when the gain is larger than the gain specified in abserr. Usually, *relerr* is on the order of about 0.01, which sets a 1% error in the rational fit when the gain is large. Usually, *abserr* is in the order of 1e-3 to 1e-4, which specifies that when the gain is less than *abserr*, allow a lot of error in the rational fit. 1e-3 is -60dB gain, and 1e-4 is -80 dB gain. The *abserr* parameter lowers the order of the model by allowing fewer poles and zeros when the gain is small.

Consider the data shown by the solid line in Figure 8-19. In this example:

- *relerr* can control the error in the passband
- *abserr* can control the error in the stopband

If you are not interested in the details of the filter behavior in the stopband, you can set *abserr* to the level shown and these details are ignored in the interpolation.

© 1994-2020 All Rights Reserved.

Figure 8-19 Using relerr and abserr

This example has approximately a 10% ripple in the passband. To ensure adequate error control there, set *relerr* to much less than 0.1, for example, 0.01.

The dashed line in Figure 8-19 shows an interpolation function with the error control levels specified in Figure 8-19: *abserr*=2e^{-4}, and *relerr*=0.01. Details of the data below the *abserr* threshold are not resolved by the rational interpolation.

To recover the details in the stopband, set *abserr* to about 1e^{-5} for this example.

If there is noise in the data, set *relerr* and *abserr* above the respective noise levels. Otherwise, Spectre and SpectreRF attempt to interpolate the noise, resulting in very high order models and very long simulation times.

Remember that if you set *relerr* to zero, then, from the formula above, pure absolute error is used. Conversely, if you set *abserr* to zero, the error control is based solely on the errors relative to the magnitude of the input data.

© 1994-2020 All Rights Reserved.

Using the rational order (ratorder) Parameter

In general, it is usually best to specify only the accuracy parameters, *relerr* and *abserr*, and to let SpectreRF automatically select *rational order,* the order of the rational approximation. However, if you have information about your data set, you can direct SpectreRF to use a specific order of approximation in the rational interpolation. For example, if you know that your tabulated S-parameter data represents a sixth-order filter, you might instruct SpectreRF to use a seventh or eighth order fit. The slightly higher order gives SpectreRF flexibility to adjust for any noise or non-ideal behavior in the data.

When you specify the *ratorder* parameter, SpectreRF tries to meet the error requirements specified by *relerr* and *abserr*, but is limited to the number of poles and zeros it can use to achieve them. If the error specified cannot be achieved with the number of poles and zeros you have specified, a warning message is generated to let you know that the error requirements were not met.

ROM data file

If this file is specified, when the rational fit is performed, the results of the rational fit are stored in the specified file as a Spectre format reduced order model.

Troubleshooting

Certain S-parameter data sets might cause difficulty for the rational interpolation process. Types of data to avoid are:

- Data specified only over a very narrow frequency range. Time-domain simulation requires time-domain models with a wider range of values; values that lie outside the narrow frequency range. Data extrapolation might be difficult and is always risky.
- Very noisy data. Noisy data might lead to large, unreliable time-domain models.
- Data on a very sparse frequency grid. Accurate interpolation of such data might be impossible.
- Data with long ideal delays.
- Data representing idealized lossless elements, such as lossless transmission lines.

© 1994-2020 All Rights Reserved.

Assessing the Quality of the Rational Interpolation

If you suspect a problem with the rational interpolation process, you can investigate it using the *sp* analysis using the following steps:

1. Construct a test schematic consisting of an *nport* component with interpolation set to rational, as discussed in "Using the nport Component" on page 1871.

2. Next add the appropriate number of *port* components to the schematic. An example with two ports is shown below.

3. Perform an *sp* analysis on the *nport* component and look for anomalies.

© 1994-2020
All Rights Reserved.

Large swings in interpolated values and S-parameter magnitudes greater than one both suggest a problem.

Large changes in the interpolated data result from an inaccurate fit. Usually, when this happens, there are local peaks or suckouts in the S-Parameter data.

S-parameter values greater than one result from a non-passive (energy-generating) model that might create unstable time-domain solutions. Be particularly critical of anomalies near the zero frequency (DC). Most S-Parameter files do not have zero frequency data, so the DC point must be extrapolated. Sometimes in rational fit, the DC point can be inaccurate.

When the anomalous behavior occurs *within* the frequency range of the data in the S-parameter data file, it usually indicates an inaccurate rational interpolation. Anomalous behavior below the first frequency point and above the last frequency point are uncommon, but always needs to be checked. If you have this anomalous behavior, linear interpolation is suggested.

Possible Problems with Rational Fit

Because the rational fit can have unexpected behavior, it is suggested that you run an S-Parameter simulation over a very wide frequency range. Check for those behaviors in the sp analysis.

© 1994-2020 All Rights Reserved.

A comparison between rational and linear interpolation is shown below. Note that S21 and S11 both go considerably above unity gain (0dB) above about 60GHz for the rational fit. The data in this file stops at 25GHz. This data is from a passive structure.

Magenta and green are the original data. Blue and red are the rational fit.

© 1994-2020
All Rights Reserved.

Also note the differences in the low frequency data shown below.

1. Verify that all the conditions listed at the beginning of the Troubleshooting section are met.
2. Try decreasing *relerr* or *abserr* or both.
3. Try specifying a higher-order interpolation with the *ratorder* parameter.
4. Remember that measured data can contain fine details that might require a higher order than you might expect from a casual inspection of the data.
5. If the anomalous behavior occurs *outside* the tabulated frequency range, try changing the *abserr*, *relerr*, or *ratorder* parameters. Sometimes anomalies can be removed by using a more accurate fit. However, there are limits to the ability to extrapolate outside the frequency interval you specify. You might need to specify additional data points in the S-Parameter file to fix the problem.

© 1994-2020 All Rights Reserved.

Difficulty can also sometimes be seen in running the transient analysis. Below is a *Choosing Analyses* form that shows a setup to run the transient to 10nsec.

© 1994-2020
All Rights Reserved.

The simulation runs, and the data plots, but note that only part of the waveform has been calculated.

When you look in the Spectre output log, the iterations have stopped because the solution goes above 1GigaVolt.

Model Reuse

Rational Fit

When *rational* or *bbspice* is set for the interpolation method, Spectre caches the rational model in the simulation directory. As long as you do not change the S-Parameter file, the rational fit will be reused.

© 1994-2020 All Rights Reserved.

Linear and Spline Fit

Prior to MMSIM 12.1, every instance of every nport calculated the impulse response at the beginning of the simulation. In some cases, this added a lot of time in order to accomplish this. Starting with MMSIM 12.1, only the impulse responses that need to be calculated are performed. For example, if there are multiple instances of nport with the same S-Parameter file and the same nport settings, the impulse response is calculated only once.

The impulse response data is cached by default so that as long as the S-Parameter data and the properties on the nport do not change, the cached data is reused in subsequent simulations. When the properties on the nport instance are modified, then a new impulse response is needed. The impulse response is recalculated and cached without eliminating the original cached data file. This is done so that if you go back to the original settings, you do not need to re-run the impulse response.

Additionally, file locking has been implemented to allow multiple users to have access to a shared impulse response directory. This is controlled by the *nportfiledir* option described below.

© 1994-2020 All Rights Reserved.

There are two options that control caching. These options are located in the ADE Explorer options located in *Simulation-Options-Analog*.

Nportirreuse

This option has the default of *yes*. Keep this at the default to save time. This option enables caching of the impulse response.

nportirfiledir

This controls the location of the cached impulse response files. The default (blank) sets the location to `/home/<username>/.cadence/mmsim`. When a directory name is specified in

© 1994-2020 All Rights Reserved.

the *nportirfiledir* option, the impulse response files are cached in the specified directory. This can be a shared directory between users, as long as all the users have write permissions in that directory.

Dcblock, dcfeed, indq, and capq

Ideal components are provided to allow what-if analysis without needing details of specific structures to allow the determination of what is needed to meet the specifications.

These components are provided for any analysis in Spectre except shooting. This allows the use of inductors and capacitors with Q in s-parameter, ac, and hbac analyses, along with the other small-signal Spectre and SpectreRF analyses.

Dcfeed

Dcfeed is an ideal short at DC and an ideal open at all other frequencies. For the transient analysis, dcfeed is an inductor with a default value of 1uh.

© 1994-2020 All Rights Reserved.

The symbol and the properties form are shown below. The *Model Name* field should be left blank.

An example comparing the dcfeed, and inductor, and a wire is shown below.

© 1994-2020
All Rights Reserved.

This example shows three cascode amplifiers with different connections to the power supply for the upper transistor. The left amplifier has a dcfeed. The center amplifier has an inductor. The right amplifier has a wire. In the amplifiers above, the resistor to the left has the output point set to 1K ohm.

When the inductor and the dcfeed are essentially open, the gain of the circuit is set by the resistor. When the impedance is less than 1Kohms, the load impedance is dominated by the impedance of the DCfeed or inductor. In this way, we can deduce what is happening at low frequency by running a simulation.

The amplitude of the input is deliberately large so the amplifiers tend toward the mixer behavior. The input signals are at 2.45GHz and 2.451GHz. The difference frequency is 1MHz. The 1 MHz signal will be analyzed so the behavior of the dcfeed, inductor, and wire can be evaluated.

At all frequencies other than DC, the dcfeed is open. The inductor impedance goes up with frequency. At 1MHz, the impedance of the 1uh inductor is 6.28 ohms.

Below is the simulation result from harmonic balance near 2.45GHz. The result agrees for the dcfeed and inductor. With a wire, essentially no signal is present.

value(dbm(spectralPower((v("/Out1...

	Expression	Value	Expression	Value	Expression	Value
1	value(dbm(spe...	9.719	value(dbm(spe...	10.01	value(dbm(spe...	-302.2

In the table above, the left result is the output signal level in dBm for the amplifier with the dcfeed, the middle is the signal level in dBm for the amplifier with the inductor, and the right is the output signal level in dBm for a wire connected to the power supply. The third output is at the numerical noise floor of the simulation.

© 1994-2020 All Rights Reserved.

Below is the spectral plot near 2.4GHz.

Below is the simulation result near 1MHz.

	Expression	Value	Expression	Value	Expression	Value
1	value(dbm(spe...	-20.74	value(dbm(spe...	-64.83	value(dbm(spe...	-629.6

In the table above, the left result comes from the amplifier with the dcfeed. The center is with the inductor, and the right is with the wire. The dcfeed is open at 1MHz, so the signal level is set by the 1K resistor. Because the inductor has an impedance of only 6.28 ohms at 1MHz, the signal is much smaller than for the dcdfeed. With a wire, you see the numerical noise floor of the simulation.

© 1994-2020 All Rights Reserved.

The spectral plot is shown below.

Dcblock

Dcblock is a near open (1Gohm) at DC and an ideal short at all other frequencies. For the transient analysis, dcfeed is a capacitor with a default value of 1uF.

© 1994-2020 All Rights Reserved.

The schematic symbol and properties list are shown below.

Below is an example with three cascode amplifiers that use different connections to the output load resistor. The left amplifier has a dcblock in series with the load resistor. The middle

© 1994-2020
All Rights Reserved.

amplifier has a capacitor in series with the load resistor. The amplifier on the right has a wire connected to the load resistor. The output resistance is 10KOhms.

Here is a table with the DC values from the three cases. The left column is the DC voltage on the load resistor with the dcblock, the center column is the DC voltage with the capacitor, and the right column is the DC voltage with the wire.

value(v("/Out4" ?result "hb_mt_fi") ...

	Expression	Value	Expression	Value	Expression	Value
1	value(v("/Out4...	28.99E-6	value(v("/Out5...	0.000	value(v("/Out6...	2.842

The voltage with the dcblock is only a few microvolts. This is because it is a 1Gohm resistor at DC. The capacitor is a perfect open at DC, so the voltage is exactly zero. With a wire, the DC bias level appears at the load resistor.

Here, is the simulation results from harmonic balance. Instead of dB, the vertical axis is linear volts. Note that the DC level is small for the dcblock. This is the top trace. The dcblock is a

© 1994-2020 All Rights Reserved.

1e9 ohm resistor at dc, and an ideal short at all non-zero frequencies. The center trace is from the amplifier with the capacitor, and the bottom trace is from the circuit with the wire.

indq

Indq is a series inductor and resistor. There are three different modes with different frequency behavior in each mode. Below is the test circuit for the plots that will come later. PORT1 is

© 1994-2020 All Rights Reserved.

defined as port1 in a single port S-parameter analysis so that the Z-Parameters for the inductor can be directly plotted.

© 1994-2020
All Rights Reserved.

The properties list for *mode 1* is shown below.

The inductor with Q works in all the analyses except for shooting.

The `indq` component has three modes, with different behavior in each mode. For all three modes, you need to specify the inductance, Q, and the frequency for the inductance and Q specification.

Mode 1

In this mode, the `indq` model is a series RL branch with R=constant and L=constant at the value given by the *Inductance* property. The value of the property *Rdc in mode2 and 3* is not used in this mode.

© 1994-2020 All Rights Reserved.

Special case: specifying Q=0 or F=0 makes the `indq` component just an inductor with the value L without any series resistance.

The resistance is calculated as follows:

$$R = \frac{\omega_Q L}{Q}$$

Below is a plot of the inductance, resistance, and Q with respect to frequency. A vertical marker has been placed at 1GHz, which is the frequency for L and Q. The inductance and resistance are constant, and the Q rises with frequency.

© 1994-2020
All Rights Reserved.

Mode 2

In this mode, both the resistance and the inductance are frequency dependent.

$$R(\omega) = R_{DC} + R'' \sqrt{\omega}$$

$$L(\omega) = L\infty + R'' \sqrt{\frac{1}{\omega}}$$

If R_{DC} in *mode 2 and 3* property is set to zero or is left blank, the DC resistance is 0.05*the resistor value that is set at the frequency defined by the *Frequency for L and Q* property.

R_{DC}, R'', and $L(\infty)$being constants are the classic forms of the skin-effect expressions for the resistance (grows as the square root of frequency) and inductance (the internal inductance decays inversely proportional to the square root of frequency). These expressions specify a causal frequency dependence of impedance. At the frequency specified in the *Frequency for L and Q property*, L and Q are equal to the values specified for *Inductance* and *Quality factor*.

The transient analysis cannot use frequency-dependent impedances, so it uses the *inductance* specified in the properties form, and the R_{DC} *in Mode 2 and 3* property for the resistance.

© 1994-2020 All Rights Reserved.

Below is the property form for mode 2.

© 1994-2020
All Rights Reserved.

Below is the inductance, resistance, and Q. Note the log frequency for inductance.

Mode 3

The advantage of mode 3 is that the non-physical frequency dependence of R near f=0 (DC) present with mode=2 is corrected. In this mode, R(f) has a reasonable frequency derivative near f=0. However, the expressions are non-causal as R varies as a function of frequency while L remains constant.

In this mode, L is constant at the value specified and

$$R = \sqrt{(R_{DC})^2 + \left(\frac{\omega L}{Q}\right)^2}$$

If the R_{DC} in *mode 2 and 3* property is set to zero, the DC resistance is 0.05*the resistor value that is set at the frequency defined by the *Frequency for L and Q* property.

Because the transient analysis cannot handle frequency-dependent resistance, the transient analysis uses the *inductance* specified in the properties form, and R_{DC} for the resistance.

© 1994-2020 All Rights Reserved.

The properties list for mode 3 is shown below.

© 1994-2020
All Rights Reserved.

The inductance, resistance, and Q in mode 3 are plotted versus frequency, as shown below.

© 1994-2020
All Rights Reserved.

Here is an expanded view of Q at low frequency. Q drops off at low frequency.

capq

Capq is a parallel capacitor and resistor. There are three different modes with different behavior in each mode. In all three modes, the capacitor is constant at the value specified in the properties form. Below is the test circuit for the plots that will come later. PORT1 is defined

© 1994-2020 All Rights Reserved.

as port1 in a single port S-parameter analysis so that the Y-Parameters for the capacitor can be directly plotted.

Mode 1

In this mode, the capq model is a parallel R C branch with R=constant and C=constant at the value given by the *Capacitance* property.

The conductance is constant as given below.

$$Real(Y) = \frac{C\omega_Q}{Q}$$

ω_Q above is the frequency for the Q value in radians per second.

© 1994-2020 All Rights Reserved.

This model is used in all the analyses, except shooting, where it is not supported.

Below is the properties list for the capq with mode 1 selected.

© 1994-2020
All Rights Reserved.

Below is a plot of the conductance, and Q with respect to frequency in mode 1. A vertical marker has been placed at 1GHz, which is the *frequency for Q*. The capacitance and resistance is constant, and the Q rises linearly with frequency.

Mode 2

In mode 2, the conductance is frequency dependent as shown in the equation below.

$$Real(Y) = \frac{C\sqrt{\omega \times \omega_Q}}{Q}$$

The capacitance is constant in all modes.

ω_Q above is the frequency where Q is specified in radians per second.

Mode 2 is non-causal. Because the resistance is frequency dependent, the transient analysis uses the capacitance specified in the properties list with no resistor in parallel.

© 1994-2020 All Rights Reserved.

Below is the properties form for mode 2.

Below is the conductance and Q versus frequency. Vertical markers have been placed at 1GHz and 40GHz. Mode 2 is a compromise between a linear Q increase with frequency in

© 1994-2020
All Rights Reserved.

mode 1, and a constant Q in mode 3. The Q goes up with frequency, but not as fast as in mode 1.

Mode 3

In mode 3, C is constant at the value specified in the properties form, and the conductance is

$$Real(Y) = \frac{\omega C}{Q}$$

Mode 3 is non-causal. Because the resistance is a function of frequency, the transient analysis uses just the capacitance term.

© 1994-2020
All Rights Reserved.

The properties list for mode 3 is shown below.

© 1994-2020
All Rights Reserved.

The conductance and Q versus frequency is plotted below.

In mode 3, the capacitance is constant, and the Q is constant. The resistance is a function of frequency.

Reference: S-Parameter Equations

This section describes the equations used by the VIVA Calculator to plot data generated by Spectre S-parameter simulations.

Using the VIVA Calculator in the Analog Design Environment, you can plot the following S-parameter data:

- Network Parameters
- Two-Port Scalar Quantities
- Two-Port Gain Quantities
- Two-Port Network Circles
- Equation for VSWR (Voltage Standing Wave Ratio)

© 1994-2020 All Rights Reserved.

- Equation for GD (group delay)

Use the buttons located in the *Function* area of the S-parameter *Direct Plot Form* to specify the type of data to plot.

Network Parameters

You can plot S, Y, Z, and H network parameters.

S-parameters, Y-parameters, and Z-parameters (denoted as SP, YP, and ZP on the *Direct Plot Form*) are defined for circuits with any number of ports. H-parameters (denoted as HP on the *Direct Plot Form*) are defined only for two-port circuits.

You can plot parameters on polar charts, Smith charts, or on rectangular plots after applying a *Modifier* option. The dB conversion uses $20 \log_{10} X$ because the parameters represent scalar ratios (for example, voltage).

Equations for Network Parameters

For the ZP, YP, and HP parameters, Spectre returns S-parameters to the analog design environment. The environment converts them, as needed, to the equivalent Z, Y, and H matrixes using standard published methods. Spectre calculates S-parameter values.

SP (S-parameter) values

SP (S-parameter) values are calculated by Spectre.

ZP (Z-parameter) equation

The Z-parameter equation is as follows:

$$Z_m = [Z_{ref}][I + S_m][I - S_m]^{-1}[Z_{ref}]$$

Where

- S_m is the N-port S-parameter matrix
- I is the N x N identity matrix
- Z_{ref} is the characteristic impedance of the port
- Z_m is the resulting Z-parameter matrix

© 1994-2020 All Rights Reserved.

YP (Y-parameter) equations

The Y-parameter equations are as follows:

$$Y_m = [Y_{ref}][I - S_m][I + S_m]^{-1}[Y_{ref}]$$

Where

- S_m is the N-port S-parameter matrix
- I is the N x N identity matrix
- Y_{ref} is a diagonal matrix defined as

$$Y_{ref} = \frac{1}{\sqrt{\Re e\{Z_i\}}}$$

Where

- Z_i is the terminating impedance at port i
- Y_m is the resulting Y-parameter matrix

The HP (H-parameter) equations

The HP (H-parameter) equations only apply to two-port circuits.

D is

$$D = (1 - S_{11})(1 + S_{22}) + S_{21}S_{12}$$

H_{11} is

$$H_{11} = \frac{[(1 + S_{11})(1 + S_{22}) - S_{21}S_{12}]Z_{ref1}^{2}}{D}$$

H_{21} is

© 1994-2020 All Rights Reserved.

$$H_{21} = \left(\frac{-2S_{21}}{D}\right)\frac{Z_{ref1}}{Z_{ref2}}$$

H_{12} is

$$H_{12} = \left(\frac{2S_{21}}{D}\right)\frac{Z_{ref1}}{Z_{ref2}}$$

and H_{22} is

$$H_{22} = \frac{(1-S_{11})(1-S_{22})-S_{21}S_{12}Z_{ref}^{2}}{D}$$

Where

- Z_{ref1} is the terminating impedance at port 1
- Z_{ref2} is the terminating impedance at port 2

Two-Port Scalar Quantities

The two-port scalar quantities include

- NF_{min} is minimum noise figure
- NF is noise figure
- R_n is equivalent noise resistance
- G_{min} is Optimum Noise reflection coefficient
- B_{1f} is alternative stability factor
- K_f is stability factor

You can plot two-port scalar quantities only against frequency. In addition, you can plot them only on rectangular charts. Figure 8-20 on page 1929 illustrates a generic two-port circuit that defines impedances and reflection coefficients.

© 1994-2020 All Rights Reserved.

Figure 8-20 A Generic Two-Port Circuit

In Figure 8-20:

- Z_s is the source impedance
- Γ_s is the input reflection coefficient
- Z_L is the load impedance
- Γ_L is the load reflection coefficient
- Z_O is the characteristic impedance

Equations for Two-Port Scalar Quantities

G_{min} (Optimum Noise Reflection Coefficient) Equation

G_{min} (also known as Γ_{min} in the literature) is the reflection coefficient associated with minimum noise figure. You can plot G_{min} on a Smith chart or, by using the *Modifier* field, on a rectangular chart.

$$G_{min} = \Gamma_{min} = \frac{Z_{on} - Z_o}{Z_{on} + Z_o}$$

Where

- Z_o is the characteristic impedance
- Z_{on} is the source impedance associated with minimum noise figure (NF_{min})

© 1994-2020 All Rights Reserved.

NF_{min} (Minimum Noise Figure) and NF (Noise Figure) Equations

The NF_{min} and NF plots are controlled by the *Modifier* option in the *Direct Plot Form*.

Plot noise figure (NF) in dB by setting *Modifier* to *dB10*

- Plot noise factor (F) by setting *Magnitude*

Use NF_{min} and NF only for two-port circuits.

$$NF = 10\log_{10}F$$

Where

- F is the noise factor
- NF_{min} is the minimum noise figure

The NF_{min} values are calculated by Spectre

The NF (noise figure) equation is calculated by the Analog Design Environment from NF_{min}, G_{min} (Γ_{min}), and r_n. You can specify the optional source reflection coefficient Γ_S as an argument if you use the Analog Design Environment calculator. From the *Direct Plot Form*, the Analog Design Environment assumes Γ_S to be 0 (input matched to reference termination).

$$NF = NF_{min} + \frac{4r_n\left|r_s - r_{min}\right|^2}{\left(1 - \left|r_s\right|^2\left|1 + r_{min}\right|^2\right)}$$

Where

$$r_n = \frac{R_n}{Z_o}$$

Here r_n is the *normalized* equivalent noise resistance.

Rn (equivalent noise resistance)

R_n plots equivalent noise resistance. The R_n values are calculated by Spectre.

© 1994-2020 All Rights Reserved.

Stability Factors

K_f and B_{1f} plot the Rollet stability factor and its intermediate term. Use these parameters only for two-port circuits.

D is

$$D = S_{11}S_{22} - S_{21}S_{12}$$

K_f is

$$K_f = \frac{1 - |S_{11}|^2 - |S_{22}|^2 + |D|^2}{2|S_{21}||S_{12}|}$$

and B_{1f} is

$$B_{1f} = 1 + |S11|^2 - |S22|^2 - |D|^2$$

Two-Port Gain Quantities

The following gain quantities are valid only for two-port circuits:

- G_A (available gain) is the power gain obtained by optimally matching the output of the network.
- G_P (power gain) is the power gain obtained by optimally matching the input of the network.
- G_T (transducer gain) shows the insertion effect of a two-port circuit. This quantity is used in amplifier design.
- G_{umx} (maximum unilateral transducer power gain)
- G_{max} (maximum available gain) shows the transducer power gain when there exists a simultaneous conjugate match at both ports.
- G_{msg} (maximum stable gain) shows the gain that can be achieved by resistively loading the two-port such that $k = 1$ and then simultaneously conjugately matching the input and

© 1994-2020 All Rights Reserved.

output ports. For conditionally stable two-ports, you can approach the maximum stable gain as you reduce the input and output mismatch. If you attempt a simultaneous conjugate match and $k < 1$, the two-port oscillates.

Equations for Two-Port Gain Calculations

G_A (Available Gain) Equations

Available gain equations for output conjugately matched.

$$G_A = \frac{|S_{21}|^2\left(1 - |\Gamma_S|^2\right)}{|1 - S_{11}\Gamma_S|^2\left(1 - |\Gamma_2|^2\right)}$$

Where

$$\Gamma_2 = S_{22} + \frac{S_{12}S_{21}\Gamma_S}{1 - S_{11}\Gamma_S}$$

Note: When you use the S-parameter *Direct Plot Form*, Γ_S is set to zero, and therefore available gain (G_A) is plotted as:

$$G_A = \frac{|S_{21}|^2}{1 - |S_{22}|^2}$$

To plot G_A for non-zero values of Γ_S, use the Analog Design Environment calculator.

G_P (Power Gain) Equations

Power gain equations for input conjugately matched.

© 1994-2020 All Rights Reserved.

$$G_P = \frac{|S_{21}|^2\left(1-|\Gamma_L|^2\right)}{|1-S_{22}\Gamma_L|^2\left(1-|\Gamma_1|^2\right)}$$

Where

$$\Gamma_1 = S_{11} + \frac{S_{12}S_{21}\Gamma_L}{1-S_{22}\Gamma_L}$$

Note: When you use the S-parameter *Direct Plot Form*, Γ_L is set to zero, and therefore the power gain, G_P, is plotted as:

$$G_P = \frac{|S_{21}|^2}{1-|S_{11}|^2}$$

To plot G_P for nonzero values of Γ_L, use the Analog Design Environment calculator.

G_T (Transducer Gain) Equations

$$G_T = \frac{\left(1-|\Gamma_S|^2\right)|S_{21}|^2\left(1-|\Gamma_L|^2\right)}{|(1-S_{11}\Gamma_S)(1-S_{22}\Gamma_L)-S_{12}S_{21}\Gamma_S\Gamma_L|^2}$$

Note: When using the S-parameter *Direct Plot Form*, the Analog Design Environment assumes that the source (Γ_S) and load (Γ_L) reflection coefficients are zero. G_T, therefore, plots the insertion gain.

$$G_T = |S_{21}|^2$$

Using the calculator, you can plot G_T and specify the source and load terminations.

© 1994-2020 All Rights Reserved.

G_{max} (Maximum Available Gain) Equations

For K >1

$$G_{max} = \left|\frac{S_{21}}{S_{12}}\right|\left[K - \sqrt{K^2 - 1}\right]$$

For K<=1

$$G_{max} = \frac{|S_{21}|}{|S_{12}|}$$

Where K is the stability factor, K_f

G_{umx} (Maximum Unilateral Transducer Power Gain) Equation

$$G_{umx} = \frac{|S_{21}|^2}{(1 - |S_{11}|^2)(1 - |S_{22}|^2)}$$

G_{msg} (Maximum Stable Power Gain) Equation

$$G_{msg} = \frac{|S_{21}|}{|S_{12}|}$$

Two-Port Network Circles

NC plots constant noise contours at the input of a two-port circuit. *GAC* plots constant gain contours at the input port, and *GPC* plots constant gain contours at the output port. Gain contour values reflect an optimum match at the opposing port.

Noise and Gain circles can be plotted at a single dB value for a range of frequencies or at a single frequency for a range of dB values. If you do not enter values for the frequency range,

© 1994-2020 All Rights Reserved.

a circle is plotted for every simulated frequency for which a circle with the specified value exists.

SSB plots stability circles at the input port, and *LSB* plots stability circles at the output port. You can also specify a limited frequency range for these contours.

Equations for Two-Port Network Circle

NC (Noise Circle) Equations

$$N_i = \frac{(F_i - F_{min})\left|1 + \Gamma_{min}\right|^2}{4r_n}$$

where

$$\Gamma_{min} = G_{min}$$

The center is calculated using

$$C_N = \frac{\Gamma_{min}}{1 + N_i}$$

The radius is calculated using

$$r_N = \sqrt{\frac{N_i^2 + N_i\left(1 - \left|\Gamma_{min}\right|^2\right)}{1 + N_i}}$$

Where i is the index number

GAC (Available Gain Circle) Equations

The center is calculated using

© 1994-2020 All Rights Reserved.

$$C_A = \frac{g_a(S_{11}{}^* - D^* S_{22})}{1 + g_a\left(\left|S_{11}\right|^2 - |D|^2\right)}$$

The radius is calculated using

$$r_A = \frac{\sqrt{1 - 2K_f\left|S_{21}S_{12}\right|g_a + \left|S_{12}S_{21}\right|^2 g_a^2}}{\left|1 + g_a\left(\left|S_{11}\right|^2 - |D|\right)^2\right|}$$

Where

$$G_a = \frac{G_A}{\left|S_{21}\right|^2}$$

And

$$D = S_{11}S_{22} - S_{12}S_{21}$$

GPC (Power Gain Circle) Equations

The center is calculated

$$C_P = \frac{g_p(S_{22}{}^* - D^* S_{11})}{1 + g_p\left(\left|S_{22}\right|^2 - |D|^2\right)}$$

The radius is calculated using

© 1994-2020 All Rights Reserved.

$$r_P = \frac{\sqrt{1 - 2K_f\left|S_{21}S_{12}\right|g_p + \left|S_{12}S_{21}\right|^2 g_p^2}}{1 + g_p\left(\left|S_{22}\right|^2 - |D|^2\right)}$$

Where

$$g_p = \frac{G_P}{\left|S_{21}\right|^2}$$

And

$$D = S_{11}S_{22} - S_{12}S_{21}$$

LSB (Load Stability Circle) Equations

The center is calculated using

$$C_L = \frac{S_{11}D^* - S_{22}^*}{|D|^2 - \left|S_{22}\right|^2}$$

The radius is calculated using

$$r_L = \left|\frac{S_{12}S_{21}}{|D|^2 - \left|S_{22}\right|^2}\right|$$

Where

© 1994-2020 All Rights Reserved.

$$D = S_{11}S_{22} - S_{12}S_{21}$$

SSB (Source Stability Circle) Equations

The center is calculated using

$$C_S = \frac{S_{22}D^* - S_{11}^*}{|D|^2 - |S_{11}|^2}$$

The radius is calculated using

$$R_S = \left| \frac{S_{12}S_{21}}{|D|^2 - |S_{11}|^2} \right|$$

Where

$$D = S_{11}S_{22} - S_{12}S_{21}$$

Equation for VSWR (Voltage Standing Wave Ratio)

VSWR is calculated from the S-parameters. You can plot the *VSWR* at any port in the circuit on a rectangular chart.

$$VSWR_i = \frac{1 + |S_{ii}|}{1 - |S_{ii}|}$$

Where i is the port number.

© 1994-2020 All Rights Reserved.

Equation for ZM (Input Impedance)

You can plot input impedance if all other ports are matched.

$$Z_m = \frac{1 + S_{ii}}{1 - S_{ii}} R$$

Where `R` the reference impedance of the port of interest and `i` is the port number.

Equation for GD (group delay)

GD (group delay) approximates the derivative of the phase with respect to frequency, normalized to 360 degrees. Units for group delay are in seconds.

$$G_d \cong \frac{-d\phi}{d\omega}$$

Group delay is calculated from the phase of the corresponding S-parameter (for example, GD_{21} corresponds to S_{21}).

References

To learn the technical details about how SpectreRF converts S-parameter data to a time-domain description, see the article “Robust rational function approximation algorithm for model generation,” by C. P. Coelho, J. R. Phillips, and L. M. Silveira. This article appeared in the proceedings of the 36th Design Automation Conference, New Orleans, LA, June 1999.

© 1994-2020 All Rights Reserved.

© 1994-2020 All Rights Reserved.

9

Cosimulation with MATLAB® and Simulink®

This chapter describes how to set up and use a cosimulation link between the MATLAB® and Simulink® system-level simulation environment and Spectre® circuit simulator RF analysis (SpectreRF). The sections in this chapter are:

- Introduction to Cosimulation with MATLAB on page 1942
- Software Requirements on page 1942
- Setting Up and Running a Cosimulation on page 1942
- Connecting the Coupler Block Into the System-Level Simulink Schematic on page 1943
- Determining How You Want to Start and Run the Cosimulation on page 1947
- Generating a Netlist for the Lower-Level Block on page 1948
- Running the Cosimulation on page 1957
- MATLAB Support Matrix on page 1959

© 1994-2020 All Rights Reserved.

Introduction to Cosimulation with MATLAB

Cosimulation combines the best of system-level simulation with lower-level analog and RF simulation. Simulink provides large libraries of DSP algorithms for generating complicated signals and post processing while SpectreRF supports transient and envelope analysis of common RF and communication circuits such as mixers, oscillators, sample and holds, and switched capacitor filters at both the transistor and behavioral levels.

Cosimulation makes it easier

- To detect concept errors early
- To detect design flaws before tape-out
- To quickly correct issues and re-simulate

The system-level design in Simulink serves as a golden reference. System designers can use the Simulink system-level design as a testbench for implementing and simulating the design. Unfortunately, when system-level designs are simulated by themselves, the effects originating from subsystems are often not considered. With cosimulation, system designers can create low-level models of critical analog blocks and use these models one at a time to analyze the performance impact of individual blocks on the system-level simulation.

Software Requirements

Cadence recommends that you use Cadence software version MMSIM 7.1 or later.

Setting Up and Running a Cosimulation

To prepare for and run a cosimulation,

1. Connect the coupler block into the system-level Simulink schematic.
2. Determine how you want to start and run the cosimulation.
3. Generate a netlist for the lower-level block that reflects how you want to start and run the cosimulation.
4. Run the cosimulation.

These steps are described in greater detail in the following sections.

Before you continue, however, be sure that the programs are ready to run.

© 1994-2020 All Rights Reserved.

➤ Add the SpectreRF and MATLAB/Simulink engine install paths to the `MATLABPATH` environment variable.

You can automate this step by adding the appropriate command to your `.cshrc` file. For example, if you are using a C shell, you can add the following command:

```
setenv MATLABPATH `cds_root spectre`/tools/spectre/simulink:${MATLABPATH}
```

Connecting the Coupler Block Into the System-Level Simulink Schematic

To prepare the Simulink part of the design for cosimulation,

1. Start MATLAB, by typing

   ```
   matlab &
   ```

   The Simulink library opens.

   This library contains the coupler module (distinguished by the Cadence logo and labeled `SpectreRF Engine`). You can insert the coupler module in any Simulink design by dragging and dropping it from this library.

2. Open your testbench or high-level design.

3. Drag and drop the `SpectreRF Engine` coupler block into the testbench and place it in the design.

4. Connect the `SpectreRF Engine` block into the design.

© 1994-2020 All Rights Reserved.

To make a signal connection, move the mouse pointer over the module pin. The pointer changes to a cross. Press the left mouse button, move the cursor to the destination pin and release the button.

5. Double-click the `SpectreRF Engine` block.

   This opens the Block Parameters: SpectreRF Engine form where you can edit the coupler block's parameter values.

The fields have the following meanings:

Field	Meaning
Number of input pins	The number of input pins to the `SpectreRF Engine` block (0…100). The input to the block is the input signal from the Simulink design.

© 1994-2020 All Rights Reserved.

Number of output pins	The number of output pins from the `SpectreRF Engine` block (0…100). The output from the block is the signal generated by the SpectreRF simulator.
Frame size	The number of samples per frame (any positive integer or -1; default: -1). Allows you to use the coupler in a loop where frame size inheritance does not work. Simulink automatically detects conflicting frame sizes. Set this value to -1 to inherit the frame size from the input.
Sample time	Sample period of the block, in seconds (default: -1). The *Sample time* is the frame period, not the time between the samples inside the frame. Allows the coupler to have only outputs, where the Simulink coupler acts as a source and defines a *Sample time*. Also allows the coupler to have only inputs. Set this value to -1 to inherit the sample period of the connected blocks. See the MATLAB and Simulink *Help* menu for more information on *Sample time* propagation.
Socket port	The number of the service that is identified by TCP/IP. Normally, the system reverses port numbers less than 1024, so this value can range from 1024 to 65535 (default: 38520).
Simulation response timeout	Maximum time, in seconds, to wait for an answer from the SpectreRF simulator during simulation. Increase this time if the simulator requires a long calculation time for each sample or frame (default: *120*).
Socket mode	TCP or UDP. Choose UDP if frame size is less 50. TCP and UDP are two different translate protocols in TCP/IP. Without CRC, UDP is usually faster in a good net environment with small data packages and TCP is usually better for large data packages.
Spectre command	Used to run a SpectreRF simulation. This parameter enables a use mode where SpectreRF is called internally by the Simulink simulation.
Show engine port labels	If checked, the `SpectreRF Engine` shows label information.

© 1994-2020 All Rights Reserved.

6. Set the number of input pins and output pins as well as any other parameters you need to set.

7. Note the Spectre command field but do not change the value now.

   Depending on how you choose to run the cosimulation, you might need to return to this field and type in a `spectre` command.

8. Click *OK*.

   The form closes and the `SpectreRF Engine` block is updated with the correct number of pins.

9. (Optional) Make other necessary changes in the Simulink testbench window.

   For example,

   - Choose *Format – Port/Signal Displays – Signal Dimensions*.

     This switches the signal dimension display on so you can see details of framed signals.

   - Choose *View – Simulink library* to display the library browser. Double-click *Math Operations*.

     The library appears. Here you can select Simulink converters, such as a `Complex to Real-Imag` block. Drag and drop blocks into the schematic and connect them as necessary.

   Double-click any placed blocks and set appropriate values.

10. Choose *Simulation – Configuration Parameters*.

© 1994-2020 All Rights Reserved.

The Configuration Parameters form opens.

11. In the *Stop time* field, set the stop time.

12. Click *OK*.

The Configuration Parameters form closes.

13. Save the modified design by choosing *file – save as*.

This completes the necessary modifications to the Simulink testbench.

Determining How You Want to Start and Run the Cosimulation

There are three ways to run a cosimulation, after all the setup is finished.

1. You can start the two applications (SpectreRF and MATLAB) separately.

This method is appropriate if you need to be able to modify both the system-level Simulink design and the analog circuit.

© 1994-2020 All Rights Reserved.

2. You can start ADE Explorer and arrange to have MATLAB start automatically.

   This method is appropriate if you are an analog design who needs to validate a circuit with system-level design input and output.

3. You can start MATLAB and arrange to have SpectreRF start automatically.

   This method is appropriate if you are a system-level designer, because, after set up, you can use the SpectreRF Engine just like another block in Simulink.

The setup differs for each of these approaches so it is useful to decide which is most appropriate for your design before continuing.

Generating a Netlist for the Lower-Level Block

The previous section describes how to insert a coupler block into the system-level design. That defines one end of the connection but you must still establish a connection with the lower-level analog block that is simulated by SpectreRF. To do that, you insert a `cosim` statement into the netlist, either by hand or by using the ADE environment.

Preparing the Netlist When Using ADE Explorer

This section describes how to prepare a netlist for cosimulation using the Analog Design Environment (ADE).

1. Start the virtuoso tool.

   ```
   virtuoso &
   ```

   The recommended version of `virtuoso` is release IC6.1.4 or later.

2. Open the schematic view of the cell.

© 1994-2020 All Rights Reserved.

For example,

3. Open the Virtuoso Analog Design Environment from the schematic editor by choosing *Launch – ADE Explorer*.

   The Launch ADE Explorer form appears.

4. In this form, select *Open Existing View* and click *OK*.

5. Choose *Setup – Matlab/Simulink – Setting*.

   The Cosimulation Options form appears.

6. In the Cosimulation Options form, click the *Select* button located beside *Cosimulation inputs*. Switch to the schematic viewer, where you see the following information below the schematic.

   ```
   Select source instance as cosimulation inputs. Press Esc when done.
   ```

   Select the sources (which are connected to the outputs from the Simulink level of the design), then press the *Esc* key.

7. In the Cosimulation Options form, click the *Select* button beside *Cosimulation outputs*. Switch to the schematic viewer, where you see the following information below the schematic.

   ```
   Select Net/Terminal as cosimulation outputs. Press Esc when done
   ```


© 1994-2020 All Rights Reserved.

Select the outputs (which are connected to the inputs at the Simulink level of the design), then press the *Esc* key.

The Cosimulation Options form looks something like this:

8. Ensure that the value of *Cosimulation socket port* is the same as the port value of the `SpectreRF Engine` block defined in "Connecting the Coupler Block Into the System-Level Simulink Schematic" on page 1943.

9. In the Cosimulation Options form, select *Enabled*.

© 1994-2020 All Rights Reserved.

10. Examine the possible values for the *Start MATLAB* field.

Value	Behavior
now	ADE launches a MATLAB session and opens a MATLAB desktop that is no different than it would be if you launched it manually. When you use the *now* value, you must start the simulation in the Simulink design before starting the SpectreRF simulation. Using the *now* value is a good way to test whether ADE can start MATLAB and open the design successfully.
before Simulation	ADE launches an internal MATLAB session and no MATLAB desktop appears. ADE opens the specified MATLAB design and initiates the MATLAB simulation before Spectre simulation starts. ADE starts MATLAB once but cannot start it again before you close the first session. Cadence suggests using the *now* value as a test before you run with the *before Simulation* value. The *now* and *before Simulation* values share one log file, which can be opened by choosing *Setup – Matlab/ Simulink – Log file*. You can use the log file to monitor the simulation when no MATLAB desktop is visible.
no	The *MATLAB start command*, *MATLAB startup directory*, and *MATLAB design name* fields become active. When you use the *no* value, you must start the simulation in the Simulink design before starting the SpectreRF simulation.

11. Continue the process of preparing the netlist according to how you want to start the applications that run the cosimulation.

To use this starting method...	Follow the guidance in this section...
Start the two applications (SpectreRF and MATLAB) separately.	"Preparing to Start the Two Applications Separately" on page 1952
Start ADE and arrange to have MATLAB start automatically.	"Preparing to Start ADE Manually and MATLAB Automatically" on page 1952

© 1994-2020 All Rights Reserved.

To use this starting method...	Follow the guidance in this section...
Start MATLAB and arrange to have SpectreRF start automatically.	"Preparing to Start MATLAB Manually and SpectreRF Automatically" on page 1954

Preparing to Start the Two Applications Separately

1. Set the *start MATLAB* value to *no*.
2. Click *OK* to close the Cosimulation Options form.
3. Use the Choosing Analyses form to set up the analysis.

   The *Stop Time* can be any value. The SpectreRF simulator synchronizes the stop time with Simulink.

   The Choosing Analyses window looks something like this:

4. Create a netlist and make sure the Simulink `cosim` statement appears in it.

Preparing to Start ADE Manually and MATLAB Automatically

1. Set the value of the *Start MATLAB* field to *before Simulation* in the Cosimulation Options form.

© 1994-2020 All Rights Reserved.

2. Type the name of the MATLAB design into the *MATLAB design name* field.

   Such designs have an extension of `.mdl`. For example, `env_d.mdl`. The MATLAB design must be in a location that is included in the `MATLABPATH` environment variable.

3. Set the value of the *Start MATLAB* field to *now*.

   A new MATLAB application starts and helps you open the Simulink design.

4. Double-click the `SpectreRF Engine` in the design and set the parameters.

   a. Set the *Sample time*. The Sample time can be set to -1, if you are uncertain about the appropriate time to use.

   b. Set *Socket Port* to the value you set in the Cosimulation Options form.

   c. Leave the *Spectre Command* field empty.

      This field is used when you wish to start SpectreRF automatically after starting MATLAB manually.

   d. Click *OK* to close the SpectreRF Engine form.

5. In the Cosimulation Options form, set *Start MATLAB* to *before Simulation*.

6. Click *OK* to close the Cosimulation Options form.

© 1994-2020 All Rights Reserved.

7. Use the Choosing Analyses form to set up the analysis.
8. Choose *File - Save* to save the design in MATLAB. Then exit from the MATLAB that was opened by ADE Explorer.

Preparing to Start MATLAB Manually and SpectreRF Automatically

1. Set the value of the *Start MATLAB* field to *now*.

   A new MATLAB application starts and helps you open the Simulink design.

2. Double-click the `SpectreRF Engine` in the design and set the parameters.

   a. Set the *Sample time*. The Sample time can be set to -1, if you are uncertain about the appropriate time to use.

   b. Set *Socket Port* to the value you set in the Cosimulation Options form.

   c. Type a command similar to the following into the *Spectre Command* field.

   `spectre netlist_file =log spectre.out`

   The Block Parameters: SpectreRF Engine window looks something like this:

   d. Click *OK* to close the Block Parameters: SpectreRF Engine form.

© 1994-2020 All Rights Reserved.

3. In the Cosimulation Options form, set *Start MATLAB* to *no*.
4. Click *OK* to close the Cosimulation Options form.
5. Use the Choosing Analyses form to set up the analysis.
6. Choose *File - Save* to save the design in MATLAB. Then exit from the MATLAB that was opened by ADE Explorer.

Preparing the Netlist Without Using a Graphical User Interface

The steps in this section are necessary only if SpectreRF needs to start MATLAB automatically. If you are using the “Start MATLAB and arrange to have SpectreRF start automatically” starting method, ensure that there is no `cosim` statement in the netlist.

1. Edit the netlist file.

   For example,

   ```
   vi tutorial1/tutorial1.scs.
   ```

2. Add a `cosim` statement to the netlist.

   For example,

   ```
   matlab cosim server="bj2lnx20"  port=38525 inputs=["VI:wave" "VQ:wave"]
   outputs=[TX_AFE_OUT_BB_I TX_AFE_OUT_BB_Q]
   ```

   The parameters for the `cosim` statement are described below:

Table 9-1 Parameters of the cosim Statement

Parameter	Meaning
`design`	Refers to the file Simulink associates with the netlist. `design="env_d.mdl"` means the testbench whose file is `env_d.mdl`.

© 1994-2020 All Rights Reserved.

Table 9-1 Parameters of the cosim Statement, *continued*

`inputs`	Input vector to identify the flow from MATLAB to `SpectreRF Engine`. The format is `["instance_name:wave" "instance_name1:wave" …]` where `wave`, is a key word. **Note:** The sequence of instances in square brackets follows the same one as the `SpectreRF Engine` port labels. This label information can be found by double-clicking the `SpectreRF Engine` and then by checking the *Show engine port labels* at the bottom in Block Parameters of the `SpectreRF Engine`. After the inputs are set in the netlist, the sources associated with the inputs are meaningless and their parameters do not affect the simulation.
`outputs`	Output vector from `SpectreRF Engine` to MATLAB. This vector can be the net name or terminal name of interest. If net name is used, only voltage is given; if terminal name is used, only current is output. To get current in an ENVLP analysis, you must add probes in the topology.
`port`	The port on the server running MATLAB. Its value should be the same as the value in the `SpectreRF Engine` in your current MATLAB design.

`server`	The name of the machine that MATLAB starts. It can be set with the machine name or with the IP address of that machine. The accepted form is
	`server = "155.110.110.110"`
	and
	`server = "bj2lnx20"`
	`server=localhost` means that Spectre and MATLAB use the same machine.

`silent`	Tells MATLAB whether to open the window during simulation. If `silent=yes`, the testbench window is opened.
`timeout`	The period of time MATLAB spends waiting for a response from the Spectre simulator. `timeout=60` stops MATLAB if it does not receive any response.

© 1994-2020 All Rights Reserved.

3. Close and save the netlist.

   After the above steps, the netlist is ready for simulation.

Running the Cosimulation

With the coupler connected into the MATLAB design and with an appropriate netlist for the low-level design, you are ready to run the cosimulation. Note that these starting methods work only when the design is prepared as described earlier in this chapter.

To use this starting method...	Follow the guidance in this section...
Start the two applications (SpectreRF and MATLAB) separately.	"Starting the Two Applications Separately" on page 1957
Start SpectreRF and arrange to have MATLAB start automatically.	"Starting SpectreRF Manually and MATLAB Automatically" on page 1958
Start MATLAB and arrange to have SpectreRF start automatically.	"Starting MATLAB Manually and SpectreRF Automatically" on page 1958

Starting the Two Applications Separately

1. Open the high-level design or testbench in the MATLAB design window.
2. In the MATLAB design window, choose *Simulation – Start*.

   The MATLAB desktop issues the following message.

   ```
   block 'modified/SpectreRF Engine': (COSIM_OK) Waiting for incoming connection
   on port 38520, timeout: 60 sec ...
   ```

   Then quickly do step 3.

   Note: The time interval between step 2 and step 3 must be within the *Simulation response timeout* defined as Block Parameters of the `SpectreRF Engine` (by double-clicking the `SpectreRF Engine`).

3. In ADE Explorer, choose *Simulation – Netlist and Run* or enter a spectre command at the command line. For example,

   ```
   spectre tutorial1/tutorial1.scs
   ```

   The cosimulation begins. The MATLAB desktop issues a message similar to the following when simulation ends.

   ```
   block 'modified/SpectreRF Engine': (COSIM_OK) Simulation finished
   ```


© 1994-2020 All Rights Reserved.

Starting SpectreRF Manually and MATLAB Automatically

To run the cosimulation by starting SpectreRF,

- If you are using ADE Explorer, open the low-level design in the ADE window.

  a. Choose *Simulation – Netlist and Run*.

  The cosimulation begins. The MATLAB desktop issues a message similar to the following when simulation ends.

  ```
  block 'modified/SpectreRF Engine': (COSIM_OK) Simulation finished
  ```

- If you are using standalone SpectreRF,

  a. At the command line, enter a command similar to the following.

  ```
  spectre tutorial1/tutorial1.scs
  ```

  The cosimulation begins.

Starting MATLAB Manually and SpectreRF Automatically

a. Start MATLAB.

```
matlab&
```

b. Choose *Simulation – Start*.

The MATLAB desktop issues a message similar to the following.

```
block 'env_d/SpectreRF Engine': (COSIM_OK) Waiting for incoming connection
on port 38525, timeout: 60 sec ...
block 'env_d/SpectreRF Engine': (COSIM_OK) Launch Spectre with commad
'spectre tutorial2.scs =log spectre.out'”
```

The cosimulation runs.

c. After the cosimulation finishes, review the MATLAB/Simulink output, close the MATLAB desktop, and exit from MATLAB.

© 1994-2020 All Rights Reserved.

MATLAB Support Matrix

Table 9-2 MATLAB Toolbox

This matrix applies to SPECTRE19.1 base and subsequent ISR versions.

SPECTRE	MATLAB Release	Supported Platform
		Linux 64
	R2018a	Supported
SPECTRE 19.1	R2018b	Supported
	R2019a	Supported
	R2019b	Supported

Note: From SPECTRE19.1, MATLAB R2017b and older versions are no longer supported.

Table 9-3 MATLAB Cosimulation

This matrix applies to SPECTRE19.1 base and subsequent ISR versions

MATLAB versions older than MATLAB R2015a are not included in this matrix.

SPECTRE	MATLAB Release	Supported Platform
		Linux 64
	R2015a	Supported
	R2015b	Supported
	R2016a	Supported
	R2016b	Supported
SPECTRE 19.1	R2017a	Supported
	R2017b	Supported
	R2018a	Supported
	R2018b	Supported
	R2019a	Supported

© 1994-2020 All Rights Reserved.

R2019b	Supported

© 1994-2020 All Rights Reserved.

10

Top-down RF Design Methodology

This section describes a methodology for designing analog RF subsystems that fit into larger DSP systems. In particular, this section describes how to use a canonical set of top-down behavioral baseband models for exploring RF architectures in the analog design environment. These models come from the following categories in *rfLib*

- Category *top_dwnBB* contains models of common RF function blocks.
  - The default view of each model is the baseband view (called *veriloga*).
  - All models in this category also have a differential passband view (called *veriloga_PB*).

  The only exceptions are the BB_loss and VGA_BB models. They are meant only for baseband analysis and have no passband view.
- Category *top_dwnPB* contains single-ended passband versions of the baseband models.
- Category *measurement* contains the instrumentation block and baseband signal generator models used to make RF measurements. These elements are not part of an RF architecture. They simply facilitate RF measurements and diagnostics.
- Category *testbenches* contains the test circuits used in this chapter to define the model specifications in the *rfLib*. Where possible, the models are specified in terms of standard RF measurements. The most precise way to describe a measurement is with a test circuit, set up instructions, and sample measurements. The circuits in the testbenches category serve that purpose

These models provide RF designers with a fast method to map RF system specifications into detailed RF designs. The baseband models facilitate fast evaluation of candidate RF architectures specified with DSP metrics. The passband views of the baseband models provide a behavioral system testbench for checking detailed designs of individual RF system components.

Baseband models are behavioral models and all behavioral models sacrifice some accuracy for increased simulation speed. Such sacrifices are usually acceptable in architectural studies because many implementation-dependent details do not affect high level decisions. The

© 1994-2020 All Rights Reserved.

modeling approach taken in top-down design is to simulate only those effects that drive the decisions at hand.

Baseband modeling in no way replaces passband modeling. Some effects missed by equivalent baseband models can affect high level decisions. However, the application of baseband models early and passband models later minimizes the number of slow simulations needed at the lower levels of design abstraction. Baseband models help you to quickly weed out designs that would surely fail tests simulated with passband models.

The success of a modeling approach to top-down design hinges on knowing how the models fit into the design flow and knowing exactly what each modeling parameter means. This chapter has two goals:

- To describe the top-down design flow, from a modeling perspective, for baseband modeling
- To define, as clearly and concisely as possible, the parameters that specify the models

Top-Down Design of RF Systems

Ideally, the digital signal processing, or DSP, team specifies an RF subsystem that fits snugly into the DSP system. A *snug fit* means that

- The specified RF subsystem does exactly what it needs to do at the lowest possible cost
- A functional specification exists that describes requirements for the RF subsystem

In a top down design flow like the one shown in Figure 10-1 on page 1964, the DSP team writes a functional specification for an RF subsystem that has not yet been designed. The functional specification describes what the RF subsystem should and should not do without describing how to build the RF subsystem.

The functional specification supplied by the DSP team describes the RF subsystem at the highest possible level of abstraction. At this point behavioral models can be specified rather than measured. This early in the design cycle, the functional specification might well be incomplete or inconsistent. A good top-down design flow can detect problems, such as omissions and inconsistencies in the design, early in the design cycle when they are easier and less expensive to fix. Problems detected later in the design cycle can be much more costly and very difficult to resolve.

Using the functional specification supplied by the DSP team and the behavioral baseband models from *rfLib*, the RF system designers can easily explore RF architectures in the analog design environment. The baseband models facilitate fast evaluation of candidate RF architectures specified with DSP metrics. By switching to the passband views of the

© 1994-2020 All Rights Reserved.

baseband models, the RF design team maps DSP measurements to RF measurements. The passband views of the baseband models provide a behavioral system testbench for checking detailed designs for individual system components.

Using the functional specification and exploring and testing with the baseband and passband models, the RF team can efficiently create a detailed design specification that fully describes the RF subsystem. The design specification can include detailed instructions for building the RF subsystem. At this stage of the design cycle, everything that is known about the design is described at the lowest level of abstraction.

You can now extract behavioral models of a detailed design from simulated measurements. The problem remains that detailed designs usually do not exist until the project is complete. To jump directly to a detailed design implies that the design flow is bottom up. Bottom up flows are important in many projects, but not in all.

DSP and RF designers sometimes have trouble communicating through specifications because the two groups deal with different metrics. For example, DSP designers deal with *bit error rates* and *error vector magnitude statistics* whereas RF designers deal with *intercept points* and *noise figures*.

The new models described here are designed to help RF system designers in two ways.

- First, the baseband models enable RF system designers to quickly explore the RF architectural space, as specified by the DSP metrics, while letting the RF engineers specify the RF system components with RF metrics. The circuit implementations of the RF system components are easier to design and test when the components are specified with standard RF metrics.
- Second, the baseband models can be switched quickly to a passband views where the RF system model can generate end-to-end RF metrics. With end-to-end metrics, the new view can quickly simulate how the detailed design of a particular RF system component affects end-to-end performance.

Use Model for Top Down Design

The following steps outline the RF design process with focus on the early phases of the design as illustrated in Figure 10-1 on page 1964.

© 1994-2020 All Rights Reserved.

Figure 10-1 The Top Down Design Flow and Use Model

Specify the RF Subsystem in Terms of DSP Metrics

Before you begin the RF subsystem top-down design flow, the DSP design team should completely specify the RF subsystem in terms of DSP metrics. This preliminary step distinguishes the end of the DSP design flow from the beginning of the RF top down design flow and formally hands-off the RF subsystem design specifications to the RF design team.

Explore Candidate Architectures with Baseband Models

The first step in top down RF design is to select a candidate RF architecture. An RF architecture is a set of interconnected RF function blocks that, taken together, describe how

© 1994-2020 All Rights Reserved.

a receiver or transmitter operates. You specify each function block in terms of standard RF metrics such as IP3, gain, bandwidth, and noise figure.

The models you use early in the design cycle as you explore candidate RF architectures must run fast. Each simulations can span hundreds of symbols and each symbol can easily span thousands of RF carrier cycles. The space defined by the function block specifications in each candidate architecture is far too vast to explore with slow, highly precise models. Models used for architectural exploration must quickly reduce the design space down to a size that can be explored with more precision.

The most efficient models for architectural exploration suppress the RF (IF) carrier and are called *baseband* models. In contrast, the *passband* models (introduced in the next step) do not suppress carriers.

You can use the Circuit Optimizer during architectural exploration to help balance the function block specifications for a candidate architecture. For example, you can use the Circuit Optimizer to minimize RMS EVM while ensuring that other measurements stay within acceptable limits.

When you have determined the nominal specifications for each function block, you must put tolerances around them. In the analog world *specifications without tolerances are meaningless***.** The tolerance space is usually explored with some mix of experience, feasibility, a variety of analyses, and outright arbitrary decisions.

There are several ways that you can use the baseband models to test candidate tolerances as well as to determine some tolerances analytically.

One way to test a candidate set of tolerances is to run a Monte Carlo analysis on the metric of interest, like RMS, EVM, or signal-to-noise ratio (SNR).

Another approach is to use the Circuit Optimizer *in reverse*, as a de-optimizer, to determine worst case performance.

Yet another approach is to compute each tolerance separately from a parametric plot. When you have determined all but 2 or 3 tolerances, you can use a multidimensional parametric analysis to map out the performance space and easily identify the remaining tolerances.

Switch to Passband Models and Create an RF System Testbench

The second step in top down RF design is to create a passband view of the system model.

The passband system model performs two functions:

- Confirms that the filters perform as expected.

© 1994-2020 All Rights Reserved.

- Creates an end-to-end testbench that you can use to design the individual function blocks.

For computational efficiency during system passband testing, at any one time, model one or two selected function blocks at the device level. Model all other blocks in the system behaviorally using passband models.

Derive the tolerances by performing the same Monte Carlo analysis or Circuit Optimizer analysis you used to test the function block tolerances in the first step, but this time replace the DSP metrics with end-to-end RF metrics. After you know how far the end-to-end RF metrics can vary, you can insert a device-level model of a function block into the testbench to see how close it drives the system toward violating a derived end-to-end RF specification.

Implement the Function Blocks with Active and Passive Devices

The last step in the top down design process is to implement the function blocks with device models. Because the function blocks are specified in terms of standard RF metrics, you can easily measure the modeling parameters to make sure they fall within the specified tolerances. You can also insert the measured parameters back into the baseband model of the system to check the DSP metrics, or insert the device-level model directly into the passband testbench to check the derived end-to-end RF specifications.

Baseband Modeling

A baseband model for an RF function block simulates what happens to the baseband representation of a signal as it passes through the block. A baseband model maps input baseband signal trajectories into output baseband signal trajectories. If you sample a baseband signal periodically in time and plot the samples in the complex plane, the resulting scatter plot shows the symbol constellation.

Figure 10-2 on page 1967 mathematically defines a baseband representation of a passband signal. The *i* and *q* signals are the real and imaginary parts of a complex signal that rides on the two phases of an RF carrier.

© 1994-2020 All Rights Reserved.

Figure 10-2 Baseband Representation of a Passband Signal

Baseband models simulate only what happens to the carrier fundamental. Consequently, they only account for non-linearities with odd symmetry. Non-linearities with even symmetry produce no output at the carrier fundamental; they affect the carrier fundamental only when cascaded. For example, a second order non-linearity in one block can create a DC offset at it's output. Upon passing through a subsequent block with another second order non-linearity, the DC offset can mix with the carrier to affect the output carrier fundamental. You should model cascaded blocks producing unfiltered even harmonics as a single baseband model rather than as separate baseband models cascaded together. The non-linearities that most often dominate performance have odd symmetry.

© 1994-2020 All Rights Reserved.

Example Comparing Baseband and Passband Models

The example in this section walks you through an Envelope analysis that illustrates the relationship between baseband and passband models. Following the simulation, you plot the baseband equivalent output signals as computed by the baseband and passband circuits.

The *BB_test_bench* schematic shown in Figure 10-3 on page 1968 illustrates the difference between passband and baseband modeling. This circuit is located in the *rfExamples* library.

Figure 10-3 The BB_test_bench Schematic

The *BB_test_bench* circuit shows a passband circuit (across the top of the schematic) and its baseband equivalent circuit (across the bottom of the schematic). The same baseband signals drive both circuits but only the passband circuit mixes the baseband signals up to RF. The power amplifier is not matched to either input or output impedances and both impedances are reactive.

Before you start, perform the setup procedures described in Chapter 3.

Opening the Baseband Test Bench Circuit

1. In the CIW, choose *File – Open*.

   The Open File form appears.

2. In the Open File form,

© 1994-2020 All Rights Reserved.

a. Choose *rfExamples* in the *Library Name* cyclic field. (Choose the editable copy of *rfExamples* you created as described in Chapter 3.)

b. Choose *BB_test_bench* in the *Cell Names* list box. Note that the *View Name* cyclic field displays *Schematic*.

c. The completed Open File form appears like the one below.

3. Click *OK*.

The Schematic window for the *BB_test_bench* appears.

4. In the Schematic window, choose *Tools – Analog Environment*.

© 1994-2020 All Rights Reserved.

The ADE window opens.

You can also choose *Tools – Analog Environment – Simulation* in the CIW to open the ADE window without opening the design. You can open the design later by choosing *Setup – Design* in the ADE window and choosing the *BB_test_bench* in the Choosing Design form.

Choosing Simulator Options

1. Choose *Setup – Simulator/Directory/Host* in the ADE window.

   The Simulator/Directory/Host form appears.

2. In the Simulator/Directory/Host form, specify the following:

   a. Choose *spectre* for the *Simulator*.

   b. Type the name of the project directory, if necessary.

   c. Highlight the *local*, *remote*, or *distributed* button to specify the *Host Mode*.

© 1994-2020 All Rights Reserved.

For remote simulation, type the name of the host machine and the remote directory in the appropriate fields. For distributed simulation, fill out the other fields that appear.

The completed form appears like the one below.

3. In the Simulator/Directory/Host form, click *OK*.
4. In the ADE window, choose *Outputs – Save All.*

   The Save Options form appears.

© 1994-2020
All Rights Reserved.

5. In the *Select signals to output (save)* section, be sure *allpub* is highlighted.

6. In the Save Options form, click *OK*.

Setting Up Model Libraries

1. In the ADE window, choose *Setup – Model Libraries*.

   The Model Library Setup form appears.

2. In the *Model Library File* field, type the full path to the model file including the file name, `rfModels.scs`.

3. In the Model Library Setup form, click *Add*.

© 1994-2020 All Rights Reserved.

The completed form appears like the one below.

4. In the Model Library Setup form, click *OK*.
5. In the ADE window, use *Analysis - Disable* to disable any analyses you ran previously. (Check the ADE window to verify whether or not an analysis is enabled.)

Setting Up the Envelope Analysis

1. Choose *Analyses – Choose* in the ADE window.

   The Choosing Analyses form appears.
2. In the Choosing Analyses form, click *envlp.*
   - **a.** Enter *ff* in the *Clock Name* field.
   - **b.** Enter `10u` in the *Stop Time* field.
   - **c.** In the *Output Harmonics* cyclic field, select *Number of harmonics*.
   - **d.** Enter `1` in the *Number of harmonics* field.
   - **e.** Select *moderate* for the Accuracy Defaults.

© 1994-2020 All Rights Reserved.

The correctly filled out form appears below.

3. In the Choosing Analyses form, click *OK.*

Running the Simulation

1. In the ADE window, choose *Simulation – Netlist and Run*.

   The output log file appears and displays information about the simulation as it runs.

© 1994-2020 All Rights Reserved.

Look in the CIW for a message that says the simulation completed successfully.

Plotting the Baseband Equivalent Output Signals

1. In the ADE window, choose *Results-Direct Plot-Main Form*.

   The Direct Plot form appears.

2. In the Direct Plot form, do the following:

   a. Highlight *Replace* for *Plotting Mode*.

   b. Highlight *envlp* for *Analysis*.

   c. Highlight *Voltage* for *Function*.

   d. Highlight *time* for *Sweep*.

   The *Net* selection appears in the *Select* cyclic field and the label *Description: Envelope Voltage vs Time* appears.

   e. Following the message at the bottom of the Direct Plot form

   Select Net on schematic...

   Click the `baseband_I_out` net.

© 1994-2020
All Rights Reserved.

The first trace appears in the waveform window.

3. In the Direct Plot form, do the following:

a. Highlight *Append* for *Plotting Mode*.

b. Leave *Voltage* set for *Function* and *time* set for *Sweep*.

c. Following the message at the bottom of the Direct Plot form

Select Net on schematic...

Click the `baseband_Q_out` net.

© 1994-2020
All Rights Reserved.

The second trace is added to the waveform window. Both baseband equivalent output signals for the baseband model are plotted.

4. In the Direct Plot form, do the following:

a. Leave *Append* for *Plotting Mode* and *Voltage* for *Function.*

b. Highlight *harmonic time* for *Sweep*.

c. Highlight *Real* for *Modifier*.

d. Following the message at the bottom of the form,

```
Select Harmonic Number on this form...
```

Select 1 for harmonic number.

e. Following the next message at the bottom of the form

```
Select Net on schematic...
```


© 1994-2020
All Rights Reserved.

Click the `RF_out` net.

A third trace is added to the waveform window.

5. In the Direct Plot form, do the following:

   a. Leave *Append* for *Plot Mode*, *Voltage* for *Function*, *harmonic time* for *Sweep*, and 1 for *Harmonic Number*.

   b. Highlight *Imaginary* for *Modifier*.

© 1994-2020
All Rights Reserved.

6. Following the message at the bottom of the form

   `Select Net on schematic...`

   Click the `RF_out` net again.

   A fourth trace is added to the waveform window. Both baseband equivalent output signals for the passband model are added to the plot.

In the waveform display window you should now see what at first appears to be two traces. When you look more closely, you should see that each trace is actually two traces, one nearly on top of the other, making a total of four traces.

The plot resulting from this example illustrates how well baseband modeling corresponds to the time-varying fundamental Fourier component computed by Envelope analysis and raises two questions:

- Why use baseband models when Envelope analysis gives the same results?

© 1994-2020 All Rights Reserved.

- Why not use baseband models all the time?

Running a transient analysis with only the baseband models answers the first question. If from the ADE window you deactivate the passband circuit by setting the `carrier_pb` variable to zero, disable the Envelope analysis, and set up and run a 10µs transient analysis, you observe the same baseband results, but the transient simulation runs over 100 times faster.

Examining the Envelope results answers the second question. If you look closely at both waveforms you notice that the baseband waveforms clip at a slightly lower level than the Envelope waveforms. This is because hard limiting of the carrier generates higher-than-third-order harmonics and the behavioral baseband model only simulates third order non-linearities.

rfLib Library Overview

The *rfLib* include three kinds of models to support baseband modeling:

- Instrumentation models
- Non-linear memoryless models
- Linear models with memory

The instrumentation models provide stimuli, diagnostics, and performance metrics relevant to the DSP system.

Both the linear models with memory and the non-linear memoryless models simulate the function blocks in an RF architecture and are specified in terms of common RF metrics. The RF function block models include input referenced white Gaussian noise as specified by noise figure. The *rfLib* includes models for the following RF function blocks—amplifiers, mixers, filters, and phase shifters; where *filters* includes single resistors, capacitors, and inductors.

The non-linear models simulate AM/AM conversion [1] with a third-order polynomial that saturates at the peak of the transfer curve. The polynomial is specified by the gain and either the input-referred IP3 or the output-referred 1 dB compression point. Only the non-linear baseband models simulate AM/PM conversion. AM/PM conversion [1] is an important effect that is hard, if not impossible, to simulate with passband behavioral models. Figure 10-4 on page 1981 shows the basic baseband non-linearity.

© 1994-2020 All Rights Reserved.

Figure 10-4 Basic, Baseband Non-Linearity

The linear models are the key to simulating loading effects at baseband. In RF integrated circuits, loading effects are important because it is often hard to integrate impedance matching networks. The baseband models of reactive elements differentiate our approach from the spreadsheet-based approaches to RF system design. The baseband capacitor and inductor models (*cap_BB* and *ind_BB* in *top_dwnBB*) let you simulate reactive loading effects in the time domain, where non-linearities are more naturally modeled.

The baseband models of reactive elements also play a key role in modeling filters. Most digital communications text books [1,2] explain that you can model a passband transfer function at baseband by simply frequency-shifting the transfer function. What these books do not describe is how to implement the resulting transfer function in a general circuit simulator such as Spectre® circuit simulator RF analysis (SpectreRF). The shifted transfer function usually lacks complex conjugate symmetry about zero frequency and therefore has a complex impulse response.

The first consequence of modeling RF function blocks at baseband is that all equivalent baseband models have four terminals instead of two:

© 1994-2020 All Rights Reserved.

- One set of terminals represents the in-phase signals, $i_{in}(t)$ and $i_{out}(t)$
- The other set of terminals represents the quadrature signals, $q_{in}(t)$ and $q_{out}(t)$

  Both sets of terminals are illustrated in Figure 10-2 on page 1967.

The mathematics illustrated in Figure 10-5 on page 1983 and Figure 10-6 on page 1984 summarize the ideas behind a time-varying coordinate transformation that models reactive elements at baseband. The mathematics apply to capacitors as well as inductors.

There is a well-documented but little-known electro-mechanical analogy for the derivation of the inductor baseband equivalent model. The four inductor terminals resemble the stator windings of a two-phase rotating machine with shaft speed equal to the RF carrier frequency. Modulation is mathematically analogous to the flux linking a stator winding due to currents in orthogonal rotor windings. The flux depends on the shaft angle just as a modulated signal depends on the carrier phase. Transforming the vectorial equation for $v=Ldi/dt$ to the rotor reference frame suppresses the RF carrier and introduces a *speed voltage* [3,4,5,6,7,8,9], or back electro-motive force (back EMF), that couples the differential equations.

An expression for the real current (i.e. the passband current) appears in Figure 10-6 on page 1984. The real current is modeled as the projection of a two-dimensional rotating vector onto a stationary axis, the *real* axis. The vector rotates with an angular velocity equal to the RF carrier frequency.

© 1994-2020 All Rights Reserved.

Figure 10-5 Passband Current for an Inductor

$$i_{real} = i_i(t)\cos(\omega_{rf}t) - i_q(t)\sin(\omega_{rf}t) = \mathrm{Real}[(i_i+j*i_q)*e^{j\omega_{rf}t}]$$

$$i_{imag} = i_i(t)\sin(\omega_{rf}t) + i_q(t)\cos(\omega_{rf}t) = \mathrm{Imag}[(i_i+j*i_q)*e^{j\omega_{rf}t}]$$

$$i=(i_i+j*i_q)*e^{j\omega_{rf}t}$$

$$v=(v_i+j*v_q)*e^{j\omega_{rf}t}$$

$$v=L\,di/dt$$

$$(v_i+j*v_q)\cancel{e^{j\omega_{rf}t}}=L*[d(i_i+j*i_q)/dt + j*\omega_{rf}*(i_i+j*i_q)]\cancel{e^{j\omega_{rf}t}}$$

The rotating vector also has a projection onto another stationary axis orthogonal to the real axis. In the baseband literature, the orthogonal projection is the Hilbert transform of the real signal. The constitutive relationship of the inductor, *v=Ldi/dt*, is expressed in terms of coordinates in a reference frame that rotates with the vector.

Figure 10-6 on page 1984 shows the constitutive inductor relationship between voltage and current in the rotating reference frame. Note that the trigonometric terms, the terms that slow simulation speed, are gone and the two projections are now coupled through *speed voltages*. The term speed voltage comes from the fact that the voltages depend on the angular speed of the rotating reference frame. In motor theory, that speed is the shaft speed. Speed voltage is similar to the back EMF in a motor. Because of speed voltages, baseband models of filters and reactive elements must have their carrier frequency specified. The carrier frequency is the frequency for which the baseband signals are referenced. For example, the carrier frequency for an RF filter would be the RF frequency while the carrier frequency for an IF filter would be the IF frequency.

© 1994-2020 All Rights Reserved.

Figure 10-6 Relationship Between Voltage and Current for an Inductor

The baseband counterparts of the passband filter models are built up from inductors and capacitors modeled in the rotating reference frame.

In the complex expression for $v=Ldi/dt$, if you replace d/dt with $j\omega$, you find that the impedance of the inductor changes from $jL\omega$ to $jL(\omega+\omega_{rf})$. The same holds for capacitors, which means a filter transfer function, $H(\omega)$, has a baseband equivalent equal to $H(\omega+\omega_{rf})$. This is simply the original passband transfer function shifted to the left by an amount equal to the carrier frequency. Our time domain baseband models are consistent with the text book frequency domain explanation of baseband modeling.

Use Model and Design Example

This section describes how to use the baseband models during the architectural design phase. The following examples show you how to

- Construct a baseband model for a simple receiver

© 1994-2020 All Rights Reserved.

- Use the Circuit Optimizer to balance specifications among the function blocks
- Create a passband testbench for the receiver

The design goals were chosen arbitrarily. The example is meant simply to illustrate how to use the library and is a derivative of the design found in [10]. If you find that some parameters are not specified, leave them as default values. You construct the receiver from left to right, from input to output.

Opening a New Schematic Window

1. In the CIW, choose *File – New – Cellview*.

   The Create New File form appears.

2. In the Create New File form,

   a. Choose *my_rfExamples* in the *Library Name* cyclic field. (Choose the editable copy of *rfExamples* you created as described in Chapter 3.)

   b. Enter `receiver_example` in the *Cell Name* field.

   c. Select *Composer-Schematic* in the *Tool* cyclic field. `schematic` appears in the *View Name* field.

   d. The completed form appears like the one below.

3. Click *OK*.

© 1994-2020 All Rights Reserved.

An empty Schematic window for the *receiver_example* appears.

Opening the Analog Environment

1. In the Schematic window, choose *Tools – Analog Environment*.

   The ADE window opens.

   The *Library*, *Cell*, and *View* names appear in the *Design* section of the ADE window.

2. Set the simulator options from the Simulator window as described in "Choosing Simulator Options" on page 1970.

3. Set up the model libraries from the Simulator window as described in "Setting Up Model Libraries" on page 1972.

Constructing the Baseband Model for the Receiver

Construct the receiver in the Schematic window by adding blocks from left to right, from input to output, as listed in Table 10-1.

© 1994-2020 All Rights Reserved.

Except for the resistor, ground, and port models (which come from the *analogLib*), all blocks come from the *rfLib*. Unless otherwise instructed, leave the port resistances at their default value of 50 Ohms.

Table 10-1 Blocks Used to Create the Receiver

Block Name and Reference	Element Name	Library and Category
CDMA signal source — See Adding the CDMA Signal Source	`CDMA_reverse_xmit`	From the *measurement* category in *rfLib*.
Resistor—attach to CDMA signal source	`res`	From *analogLib*
Driver — See Adding the Driver	`BB_driver`	From the *measurement* category in *rfLib*.
Low noise amplifier — See Adding the Low Noise Amplifier	`LNA_BB`	From the *top_dwnBB* category in *rfLib*.
Butterworth bandpass filter — See Adding a Butterworth Band Pass Filter	`BB_butterworth_bp`	From the *top_dwnBB* category in *rfLib*
RF-to-IF mixer — See Adding an RF-to-IF Mixer	`dwn_cnvrt`	From the *top_dwnBB* category in *rfLib*
Butterworth bandpass filter — See Adding Another Butterworth Bandpass Filter	`BB_butterworth_bp`	From the *top_dwnBB* category in *rfLib*
IQ demodulator — See Adding an IQ Demodulator	`IQ_demod_BB`	From the *top_dwnBB* category in *rfLib*
Butterworth lowpass filters (create two) — See Adding Two Butterworth Lowpass Filters	`butterworth_lp`	From the *top_dwnPB* category in *rfLib*
Instrumentation model — See Adding an Instrumentation Block	`offset_comms_instr`	From the *measurement* category in *rfLib*.
Terminator — See Adding an Instrumentation Terminator	`instr_term`	From the *measurement* category in *rfLib*
Grounds—attach to RF-to-IF mixer, IQ demodulator, and Instrumentation model	`gnd`	From *analogLib*

© 1994-2020 All Rights Reserved.

Adding the CDMA Signal Source

Add the first receiver block, a CDMA signal source (`CDMA_reverse_xmit`), to the schematic.

1. In the Schematic window, choose *Add – Instance*.

   The Add Instance form appears. It may be empty or it may display information for a previously added element. The default for the *View* field is `symbol`.

2. In the Add Instance form, click *Browse*.

   The Library Browser - Add Instance form appears.

3. In the Library Browser - Add Instance form,

   a. If necessary, click *Show Categories* to display the *Category* column so you can view the elements (or cells) in the *rfLib* by category.

   b. In the *Libraries* column, click *rfLib* to display categories of elements in *rfLib*.

   The *Everything* category is displayed by default and all cells in *rfLib* are listed in the *Cells* column. (In the Add Instance form, *rfLib* displays in the *Library* field.)

   c. In the *Category* column, click *measurement* to list only the cells in the *measurement* category.

   d. In the *Cell* column, click *CDMA_reverse_xmit*.

© 1994-2020 All Rights Reserved.

In the Library Browser, cell *CDMA_reverse_xmit* and it's default view *symbol* are both selected.

In the Add Instance form,

- `rfLib` appears in the *Library* field
- `CDMA_reverse_xmit` displays in the *Cell* field
- `symbol` displays in the *View* field

The CDF parameters for the element and their default values appear at the bottom of the form.

4. To place a *CDMA_reverse_xmit* block in the schematic,

a. Move the cursor over the Schematic window.

© 1994-2020
All Rights Reserved.

The outline for the *CDMA_reverse_xmit* symbol is attached to the cursor.

b. Move the cursor near the top left corner of the schematic and click to place the *CDMA_reverse_xmit* block.

This block models a CDMA signal source.

c. Click *Esc* to remove the symbol from the cursor.

Adding a Resistor to the CDMA Signal Source

Because this example does not use the binary output nodes (`i_bin_node` and `q_bin_node`) on the CDMA signal source, connect a resistor between these nodes to avoid unused pin warnings.

1. In the *Libraries* column of the Library Browser - Add Instance form, click *analogLib* to display elements in *analogLib*.

   If *Show Categories* is selected, the *Everything* category is displayed by default and all cells in *analogLib* are listed in the *Cells* column.

2. Scroll through the list of cells in *analogLib* to locate the resistor cell, *res*.

3. Click *res* in the *Cell* column.

© 1994-2020 All Rights Reserved.

The cell *res* and it's default view *symbol* are both selected.

In the Add Instance form,

- `analogLib` displays in the *Library* field
- `res` displays in the *Cell* field
- and `symbol` displays in the *View* field

© 1994-2020
All Rights Reserved.

The CDF parameters for the element and their default values appear at the bottom of the form.

4. Move the cursor over the Schematic window.
5. Click to place the top resistor terminal in line with the top binary output node (`i_bin_out`) on the lower right side of the *CDMA_reverse_xmit* block
6. Click *Esc* to remove the symbol from the cursor.

Wiring the Resistor to the CDMA Signal Source

Wire the resistor to the binary outputs, `i_bin_node` and `q_bin_node`, of the CDMA signal source.

1. To wire the resistor to the *CDMA_reverse_xmit* block, in the Schematic window choose *Add - Wire (narrow)*.
2. Click `i_bin_node` on the `CDMA_reverse_xmit` block then move the cursor and click the top node of the resistor.
3. Click `q_bin_node` on the `CDMA_reverse_xmit` block then move the cursor and click the bottom node of the resistor.
4. Click *Esc* to stop wiring.

© 1994-2020 All Rights Reserved.

The CDMA signal source and resistor wired together appear as follows.

Adding the Driver

Add a driver block to the right of the CDMA signal source block.

1. In the Schematic window, choose *Add – Instance* to display the Add Instance form.
2. In the Add Instance form, click *Browse* to display the Library Browser - Add Instance form.
3. In the Library Browser - Add Instance form, make the following selections.

Library	Category	Cell	View
rfLib	*measurement*	*BB_driver*	*symbol*

At the top of the Add Instance form,

- `rfLib` displays in the *Library* field,
- `BB_driver` displays in the *Cell* field and
- `symbol` displays in the *View* field.

© 1994-2020 All Rights Reserved.

- The CDF parameters and their default values appear at the bottom of the Add Instance form.

4. Move the cursor over the Schematic window.
5. Click to place the *BB_driver* to the right of the *CDMA_reverse_xmit* block. Align the input pins of the driver with the analog output pins of the CDMA signal source.
6. Click *Esc* to remove the symbol from the cursor.

Wiring the Signal Source to the Driver

1. To wire the *BB_driver* block to the *CDMA_reverse_xmit* block, in the Schematic window choose *Add – Wire (narrow)*.
2. Click `i_out_node` on the *CDMA_reverse_xmit* block then click `I_in` on the *BB_driver* block.
3. Click `q_out_node` on the *CDMA_reverse_xmit* block then click `Q_in` on the *BB_driver block*.
4. Click *Esc* to stop wiring.

© 1994-2020 All Rights Reserved.

The schematic now appears as follows.

Modifying Parameter Values for the Driver

Edit the value of the *BB_driver* CDF parameter *dBm-out@1v peak in driver* as follows.

1. Choose *Edit – Properties – Objects* in the Schematic window.

   The Edit Object Properties form appears. You use this form to change the values of CDF (component description format) properties for the driver and modify the schematic for this simulation.

2. In the Schematic window, click the *BB_driver* block.

3. The Edit Object Properties form changes to display information for the *BB_driver* block

4. Change the *dBm-out@1v peak in* parameter value as follows.

Parameter Name	Value
dBm-out@1v peak in	*-16*

© 1994-2020 All Rights Reserved.

The driver converts 1 peak volt from the CDMA signal source to –16 dBm referenced to the output resistance of the driver.

5. Click *OK* in the Edit Object Properties form.

   The form closes.

Adding the Low Noise Amplifier

Add a low noise amplifier to the right of the driver.

1. In the Schematic window, choose *Add – Instance* to display the Add Instance form.

   The Add Instance form appears. It may be empty or it may display information for a previously added element.

2. In the Add Instance form, click *Browse* to display the Library Browser - Add Instance form.

3. In the Library Browser - Add Instance form, make the following selections. (If necessary, click *Show Categories* at the top of the Library Browser, to display the *Category* column.)

Library	Category	Cell	View
rfLib	top_dwnBB	LNA_BB	symbol

In the Library Browser, cell *LNA_BB* and it's default view *symbol* are both selected.

At the top of the Add Instance form,

- `rfLib` displays in the *Library* field
- `LNA_BB` displays in the *Cell* field
- `symbol` displays in the *View* field

© 1994-2020 All Rights Reserved.

The CDF parameters for the element and their default values display at the bottom of the LNA_BB Add Instance form.

4. Move the cursor over the Schematic window. The outline for the LNA symbol is attached to the cursor. Align the input pins of the LNA with the output pins of the driver.
5. Click to place the *LNA_BB* block to the right of the *BB_driver* block.
6. Click *Esc* to remove the symbol from the cursor.

Wiring the Driver to the LNA

1. To wire the *LNA_BB* block to the *BB_driver* block, in the Schematic window choose *Add - Wire (narrow)*.
2. Click `I_out` on the *BB_driver* block then click `I_in` on the *LNA_BB* block.
3. Click `Q_out` on the *BB_driver* block then click `Q_in` on the *LNA_BB* block.
4. Click *Esc* to stop wiring.

© 1994-2020 All Rights Reserved.

The schematic now appears as follows.

Modifying Parameter Values for the LNA

1. Edit the CDF parameter values for the LNA.

   a. Choose *Edit – Properties – Objects* in the Schematic window.

   The Edit Object Properties form appears. You use this form to change the list of CDF (component description format) properties for the LNA and modify the schematic for this simulation.

   b. In the Schematic window, click the *LNA*.

   The Edit Object Properties form changes to display information for the LNA.

   c. Change the CDF parameter values for the LNA as follows.

Parameter Name	Value
Available pwr gain [dB]	`lna_gain`
Input resistance	`50`
Output resistance	`300`
Input referred IP3 [dBm]	`lna_ip3`
Noise figure [dB]	`10`

© 1994-2020 All Rights Reserved.

Parameter Name	Value
am/pm sharpness	`2`
cmp [dBm]	`lna_ip3`
|radians| @ cmp	`.05`
|radians| @ big input	`.7`
{1, 0, -1} for {cw, none, ccw}	`1`

2. Click *OK* in the Edit Object Properties form.

   The form closes.

Adding a Butterworth Band Pass Filter

Add a Butterworth band pass filter to the right of the low noise amplifier.

1. In the Schematic window, choose *Add – Instance* to display the Add Instance form.

   Information for the LNA is still displayed in the form.

2. In the Add Instance form, click *Browse* to display the Library Browser - Add Instance form.

3. In the Library Browser - Add Instance form, make the following selection.

Library	Category	Cell	View
rfLib	*top_dwnBB*	*BB_butterworth_bp*	*symbol*

In the Library Browser, cell *BB_butterworth_bp* and it's default view are both selected.

At the top of the Add Instance form,

- `rfLib` displays in the *Library* field
- `BB_butterworth_bp` displays in the *Cell* field
- `symbol` displays in the *View* field.

© 1994-2020 All Rights Reserved.

- The CDF parameters for the Butterworth band pass filter and their default values display at the bottom of the Add Instance form.

4. Move the cursor over the Schematic window. The outline for the filter symbol is attached to the cursor. Align the input pins of the filter with the output pins of the LNA.
5. Click to place the *BB_butterworth_bp* block to the right of the *LNA_BB* block.
6. Click *Esc* to remove the symbol from the cursor.

Wiring the LNA to the Filter

1. To wire the *BB_butterworth_bp* block to the *LNA_BB* block, in the Schematic window choose *Add – Wire (narrow)*.
2. Click `I_out` on the *LNA_BB* block then click `ini` on the *BB_butterworth_bp* block.
3. Click `Q_out` on the *LNA_BB* block then click `inq` on the *BB_butterworth_bp* block.
4. Click *Esc* to stop wiring.

© 1994-2020 All Rights Reserved.

The schematic now appears as follows.

Modifying Parameter Values for the Band Pass Filter

Edit the CDF parameter values for the Butterworth band pass filter.

1. Choose *Edit – Properties – Objects* in the Schematic window.

   The Edit Object Properties form appears. You use this form to change the list of CDF (component description format) properties for the filter and modify the schematic for this simulation.

2. In the Schematic window, click the filter.

   The Edit Object Properties form changes to display information for the filter.

3. Edit the parameter values to match those in Table 10-2.

4. Click *OK*.

Table 10-2 CDF Parameter Values for the Butterworth Filter

Parameter Name	Value
Filter order	`3`
Input impedance	`50`
Output impedance	`50`
Center frequency (Hz)	`frf`

© 1994-2020 All Rights Reserved.

Table 10-2 CDF Parameter Values for the Butterworth Filter

Parameter Name	Value
Relative bandwidth	`rf_rbw`
Insertion loss (dB)	`3`
Carrier frequency	`frf`

Specify the *Carrier frequency* parameter value for the baseband equivalent model of the Butterworth band pass filter, just as you do for any reactive element. As shown in Figure 10-6 on page 1984, the carrier frequency is used to compute speed voltages. Because filters are built up from inductors and capacitors which have speed voltages, you must specify the carrier frequency for filters.

When a filter follows an RF-to-IF mixer, its *Carrier frequency* parameter value is the IF frequency.

- The *Carrier frequency* is the frequency value to which the baseband signals are referenced.
- The *Center frequency* is the frequency for which a filter is designed. The *Center frequency* parameter value for a bandpass filter does not have to equal the *Carrier frequency* parameter value.

Adding an RF-to-IF Mixer

Add an RF-to-IF mixer (*dwn_cnvrt*) block to the right of the bandpass filter block.

1. In the Schematic window, choose *Add – Instance* to display the Add Instance form.

   The Add Instance form appears. It may be empty or it may display information for a previously added element.

2. In the Add Instance form, click *Browse* to display the Library Browser - Add Instance form.

3. In the Library Browser - Add Instance form, make the selections indicated in Table 10-3.

Table 10-3 Library Browser selections for the RF to IF Mixer

Library	Category	Cell	View
rfLib	*top_dwnBB*	*dwn_cnvrt*	*symbol*

© 1994-2020 All Rights Reserved.

In the Library Browser, cell *dwn_cnvrt* (the RF-to-IF mixer) and it's default view *symbol* are both selected.

At the top of the Add Instance form,

- `rfLib` displays in the *Library* field
- `dwn_cnvrt` displays in the *Cell* field
- `symbol` displays in the *View* field.

The CDF parameters for the down converter and their default values display at the bottom of the form.

4. Move the cursor over the Schematic window. The outline for the RF-to-IF Mixer symbol is attached to the cursor. Align the input pins of the mixer with the output pins of the filter.
5. Click to place the *dwn_cnvrt* block to the right of the *BB_butterworth_bp* block.
6. Click *Esc* to remove the symbol from the cursor.

© 1994-2020 All Rights Reserved.

Grounding the phase_err Pin on the Mixer

It is necessary to ground the phase error (*phase_err*) pin on the bottom of the RF-to-IF mixer.

1. In the Schematic window, choose *Add – Instance* to display the Add Instance form.

   The Add Instance form appears. It may be empty or it may display information for a previously added element.

2. In the Add Instance form, type

   - ❑ `analogLib` in the *Library* field
   - ❑ `gnd` in the *Cell* field
   - ❑ `symbol` in the *View* field.

3. Move the cursor over the Schematic window.
4. Click to place the ground terminal in line with the phase error node (`phase_err`) on the bottom of the *dwn_cnvt* block.
5. Click *Esc* to remove the symbol from the cursor.

Wiring the Filter and Ground to the Mixer

1. To wire the *dwn_cnvrt* block to the *BB_butterworth_bp* block and the ground, in the Schematic window choose *Add - Wire (narrow)*.
2. Click `outi` on the *BB_butterworth_bp* block then click `I_in` on the *dwn_cnvrt* block.
3. Click `outq` on the *BB_butterworth_bp* block then click `Q_in` on the *dwn_cnvrt* block.
4. Click the port on the *gnd* block then click `phase_err` on the *dwn_cnvrt* block.
5. Click *Esc* to stop wiring.

© 1994-2020 All Rights Reserved.

The schematic now appears as follows.

Modifying Parameter Values for the RF-to-IF Mixer

1. Edit the CDF parameter values for the RF-to-IF mixer (*dwn_cnvrt*) as listed in Table 10-4 on page 2005.

   a. Choose *Edit – Properties – Objects* in the Schematic window.

   The Edit Object Properties form appears. You use this form to change the list of CDF (component description format) properties for *dwn_cnvrt* and modify the schematic for this simulation.

   b. In the Schematic window, click *dwn_cnvrt*.

   The Edit Object Properties form changes to display information for *dwn_cnvrt*.

   c. Change the parameter values to match those in Table 10-4.

   d. Click OK.

Table 10-4 CDF Parameter Values for the RF-to-IF Mixer

Parameter Name	Value
available power gain[dB]	`if_mx_gain`
Input resistance	`50`
output resistance	`50`
input referred ip3[dBm]	`if_mx_ip`
noise figure [dB]	`10`

© 1994-2020 All Rights Reserved.

Table 10-4 CDF Parameter Values for the RF-to-IF Mixer

Parameter Name	Value
RF frequency	`frf`
LO frequency	`flo1`
AM/PM input point[dBm]	`-30`
phase shift at cmp[rad]	`.7`
phase shift at infinity	`2`
sharpness factor	`2`
{1,0,–1} = {cw,none,ccw}	`0`

Adding Another Butterworth Bandpass Filter

Add another Butterworth band pass filter block to the right of the RF-to-IF Mixer block.

1. In the Schematic window, choose *Add – Instance* to display the Add Instance form.
2. In the Add Instance form, click *Browse* to display the Library Browser - Add Instance form.
3. In the Library Browser - Add Instance form, make the following selection.

Library	Category	Cell	View
rfLib	*top_dwnBB*	*BB_butterworth_bp*	*symbol*

In the Library Browser, cell *BB_butterworth_bp* and it's default view are both selected.

At the top of the Add Instance form,

- `rfLib` displays in the *Library* field
- `BB_butterworth_bp` displays in the *Cell* field
- `symbol` displays in the *View* field.

© 1994-2020 All Rights Reserved.

The CDF parameters for the Butterworth band pass filter and their default values display at the bottom of the Add Instance form.

4. Move the cursor over the Schematic window. The outline for the filter symbol is attached to the cursor. Align the input pins of the filter with the output pins of the LNA.
5. Click to place the *BB_butterworth_bp* block to the right of the *LNA_BB* block.
6. Click *Esc* to remove the symbol from the cursor.

Wiring the Mixer to the Filter

1. To wire the *BB_butterworth_bp* block to the *dwn_cnvrt* block, in the Schematic window choose *Add - Wire (narrow)*.
2. Click `I_out` on the *dwn_cnvrt* block then click `ini` on the *BB_butterworth_bp* block.
3. Click `Q_out` on the *dwn_cnvrt* block then click `inq` on the *BB_butterworth_bp* block.
4. Click *Esc* to stop wiring.

© 1994-2020 All Rights Reserved.

The schematic now appears as follows.

Modifying Parameter Values for the Band Pass Filter

1. Edit the CDF parameter values for the Butterworth band pass filter as listed in Table 10-5 on page 2008.

   a. Choose *Edit – Properties – Objects* in the Schematic window.

   The Edit Object Properties form appears. You use this form to change the list of CDF (component description format) properties for the filter and modify the schematic for this simulation.

   b. In the Schematic window, click the filter.

   The Edit Object Properties form changes to display information for the filter.

   c. Change the parameter values to match those in Table 10-5.

   d. Click OK.

Table 10-5 CDF Parameter Values for the Second Butterworth Filter

Parameter Name	Value
Filter Order	`3`
Input impedance	`50`
Output impedance	`50`
Center frequency (Hz)	`-frf+flo1`
Relative bandwidth	`if_rbw`

© 1994-2020 All Rights Reserved.

Table 10-5 CDF Parameter Values for the Second Butterworth Filter

Parameter Name	Value
Insertion loss (dB)	`1`
Carrier frequency	`-frf+flo1`

As for the first band pass filter, specify the carrier frequency for the baseband equivalent model of the Butterworth band pass filter, just as you do for any reactive element. As shown in Figure 10-6 on page 1984, the carrier frequency is used to compute speed voltages. Because filters are built up from inductors and capacitors which have speed voltages, you must specify the carrier frequency for filters.

When a filter follows an RF-to-IF mixer, its carrier frequency is the IF frequency. The carrier frequency is the frequency to which the baseband signals are referenced. The center frequency of the bandpass filter does not have to equal the carrier frequency. The center frequency is the frequency for which a filter is designed.

Adding an IQ Demodulator

Add an IQ Demodulator (*IQ_demod_BB*) block to the right of the bandpass filter block.

1. In the Schematic window, choose *Add – Instance* to display the Add Instance form.

   The Add Instance form appears. It may be empty or it may display information for a previously added element.

2. In the Add Instance form, click *Browse* to display the Library Browser - Add Instance form.

3. In the Library Browser - Add Instance form,

   a. If necessary, click *Show Categories* to display the *Category* column.

   b. Click *rfLib* to display elements in *rfLib*. The *Everything* category is displayed by default and all cells in *rfLib* are listed in the *Cells* column.

   c. In the *Category* column, click *top_dwnBB* to display cells in the *top_dwnBB* category.

   d. In the *Cell* column, click *IQ_demod_BB*.

© 1994-2020 All Rights Reserved.

In the Library Browser, cell *IQ_demod_BB* (the IQ demodulator) and it's default view *symbol* are both selected.

© 1994-2020 All Rights Reserved.

At the top of the Add Instance form, `rfLib` displays in the *Library* field, `IQ_demod_BB` displays in the *Cell* field and `symbol` displays in the *View* field. The CDF parameters for the IQ demodulator and their default values display at the bottom of the form.

4. Move the cursor over the Schematic window. The outline for the IQ demodulator symbol is attached to the cursor. Align the input pins of the IQ demodulator with the output pins of the filter and click to place the *IQ_demod_BB* block to the right of the second *BB_butterworth_bp* block.

5. Click *Esc* to remove the symbol from the cursor.

© 1994-2020 All Rights Reserved.

Grounding the phase_err Pin on the IQ Demodulator

It is necessary to ground the phase error (*phase_err*) pin on the bottom of the IQ demodulator.

1. In the Schematic window, choose *Add – Instance* to display the Add Instance form.

   The Add Instance form appears. It may be empty or it may display information for a previously added element.

2. In the Add Instance form, type

   - `analogLib` in the *Library* field
   - `gnd` in the *Cell* field
   - `symbol` in the *View* field.

3. Move the cursor over the Schematic window.
4. Click to place the ground terminal in line with the phase error node (`phase_err`) on the bottom of the *IQ_demod_BB* block.
5. Click *Esc* to remove the symbol from the cursor.

Wiring the Filter and Ground to the IQ Demodulator

1. To wire the *IQ_demod_BB* block to the *BB_butterworth_bp* block and the ground, in the Schematic window choose *Add - Wire (narrow)*.
2. Click `outi` on the *BB_butterworth_bp* block then click `I_in` on the *IQ_demod_BB* block.
3. Click `outq` on the *BB_butterworth_bp* block then click `Q_in` on the *IQ_demod_BB* block.
4. Click the port on the *gnd* block then click `phase_err` on the *IQ_demod_BB* block.

© 1994-2020 All Rights Reserved.

5. Click *Esc* to stop wiring.

   The schematic now appears as follows.

Modifying Parameter Values for the IQ Demodulator

1. Edit the CDF parameter values for the IQ Demodulator (*IQ_demod_BB*) as listed in Table 10-6 on page 2013.

   a. Choose *Edit – Properties – Objects* in the Schematic window.

   The Edit Object Properties form appears. You use this form to change the list of CDF (component description format) properties for *IQ_demod_BB* and modify the schematic for this simulation.

   b. In the Schematic window, click *IQ_demod_BB*.

   The Edit Object Properties form changes to display information for *IQ_demod_BB*.

   c. Change the parameter values to match those in Table 10-6.

   d. Click OK.

Table 10-6 CDF Parameter Values for the IQ Demodulator

Parameter Name	Value
available I-mixer gain[dB]	0
available Q-mixer gain[dB]	0
Input resistance	50
output resistance	50

© 1994-2020 All Rights Reserved.

Table 10-6 CDF Parameter Values for the IQ Demodulator

Parameter Name	Value
I–[dBm] input referred IP3	40
Q–[dBm] input referred IP3	40
noise figure [dB]	2
quadrature error	0
I–sharpness factor	2
Q–sharpness factor	2
I_cmp	–30
Q_cmp	–30
I–radians@I_cmp	.7
Q–radians@Q_cmp	.7
I–radians@big I-input	2
Q–radians@big Q-input	2
I {1,0,–1} for {cw,none,ccw}	0
Q {1,0,–1} for {cw,none,ccw}	0

Adding Two Butterworth Lowpass Filters

1. Add two Butterworth low pass filters to the right of the IQ demodulator block.
   - Align one filter with the demodulator's `i_out` pin.
   - Align the other filter with its `q_out` pin.
2. In the Schematic window, choose *Add – Instance* to display the Add Instance form.
3. In the Add Instance form, click *Browse* to display the Library Browser - Add Instance form.
4. In the Library Browser - Add Instance form, make the following selections.

Library	Category	Cell	View
rfLib	*top_dwnPB*	*butterworth_lp*	*symbol*

© 1994-2020 All Rights Reserved.

In the Library Browser, cell *butterworth_lp* and it's default view are both selected.

At the top of the Add Instance form,

- `rfLib` displays in the *Library* field
- `butterworth_lp` displays in the *Cell* field
- `symbol` displays in the *View* field.
- The CDF parameters for the Butterworth low pass filter and their default values display at the bottom of the Add Instance form.

5. Move the cursor over the Schematic window. The outline for the butterworth low pass filter symbol is attached to the cursor.

   Align the *in* pin of the first butterworth low pass filter with the *I_out* pin (the top pin) of the IQ demodulator and click to place the filter close to the demodulator. Align the *in* pin of the second butterworth low pass filter with the *Q_out* pin (the bottom pin) of the IQ demodulator. You have to place it further from the demodulator to align it with the *Q_out* pin.

6. Click *Esc* to remove the symbol from the cursor.

Wiring the IQ Demodulator to the Filters

1. To wire the *IQ_demod_BB* block to the *butterworth_lp* blocks, in the Schematic window choose *Add - Wire (narrow)*.
2. Click `I_out` on the *IQ_demod_BB* block then click `in` on the first *butterworth_lp* block.
3. Click `Q_out` on the *IQ_demod_BB* block then click `in` on the second *butterworth_lp* block.
4. Click *Esc* to stop wiring.

© 1994-2020 All Rights Reserved.

The schematic now appears as follows.

Modifying Parameter Values for Both Low Pass Filters

1. Edit the CDF parameter values for the Butterworth low pass filters as listed in Table 10-7 on page 2016.

   a. Choose *Edit – Properties – Objects* in the Schematic window.

   The Edit Object Properties form appears. You use this form to change the list of CDF (component description format) properties for each filter and modify the schematic for this simulation.

   b. In the Schematic window, click the first low pass filter.

   The Edit Object Properties form changes to display information for the filter.

   c. Change the parameter values to match those in Table 10-7.

Table 10-7 CDF Parameter Values for the Butterworth Low Pass Filters

Parameter Name	Value
Filter Order	3
Input impedance	50
Output impedance	50
Corner frequency (Hz)	10M
Insertion loss (dB)	0

   d. Click Apply.

© 1994-2020 All Rights Reserved.

e. In the Schematic window, click the second low pass filter.

The Edit Object Properties form displays the information you entered for the filter as shown in Table 10-7.

f. Click OK.

Adding an Instrumentation Block

Add an instrumentation block (*offset_comms_instr*) block to the right of the low pass filter blocks.

1. In the Schematic window, choose *Add – Instance* to display the Add Instance form.

   The Add Instance form appears. It may be empty or it may display information for a previously added element.

2. In the Add Instance form, click *Browse* to display the Library Browser - Add Instance form.

3. In the Library Browser - Add Instance form, make the following selections. (If necessary, click Show Categories at the top of the Library Browser, to display the Category column.)

Library	Category	Cell	View
rfLib	*measurement*	*offset_comms_instr*	*symbol*

In the Library Browser, cell *offset_comms_instr* and it's default view *symbol* are both selected.

At the top of the Add Instance form,

- `rfLib` displays in the *Library*
- `offset_comms_instr` displays in the *Cell* field
- `symbol` displays in the *View* field

© 1994-2020 All Rights Reserved.

The CDF parameters for the *offset_comms_instr* and their default values display at the bottom of the form.

4. Move the cursor over the Schematic window. The outline for the *offset_comms_instr* block symbol is attached to the cursor. Align the *I_in* and *Q_in* pins of the *offset_comms_instr* block with the *out* pins of the butterworth low pass filters and click to place the *offset_comms_instr* block to the right of the low pass filter blocks.
5. Click *Esc* to remove the symbol from the cursor.

Grounding the Reference Pins on the Instrumentation Block

It is necessary to ground the reference pins (*I_ref* and *Q_ref*) pin near the lower left corner of the instrumentation block.

© 1994-2020 All Rights Reserved.

Grounding the phase_err Pin on the Mixer

1. In the Schematic window, choose *Add – Instance* to display the Add Instance form.

   The Add Instance form appears. It may be empty or it may display information for a previously added element.

2. In the Add Instance form, type

   - ❑ `analogLib` in the *Library* field
   - ❑ `gnd` in the *Cell* field
   - ❑ `symbol` in the *View* field.

3. Move the cursor over the Schematic window.
4. Click to place the ground terminal in line with the `Q_ref` node on the bottom of the *offset_comms_instr* block.
5. Click *Esc* to remove the symbol from the cursor.

Wiring the Filter and Ground to the Instrumentation Block

1. To wire the low pass filters to the *offset_comms_instr* block and the ground, in the Schematic window choose *Add - Wire (narrow)*.
2. Click `out` on the upper *butterworth_lp* block (the low pass filter connected to the `I_out` node on the IQ demodulator) then click `I_in` on the *offset_comms_instr* block.
3. Click `out` on the lower *butterworth_lp* block (the low pass filter connected to the `Q_out` node on the IQ demodulator) then click `Q_in` on the *offset_comms_instr* block.
4. Click the port on the *gnd* block then click the `Q_ref` node on the *offset_comms_instr* block.
5. Click the port on the *gnd* block then click the `I_ref` node on the *offset_comms_instr* block.
6. Click *Esc* to stop wiring.

© 1994-2020 All Rights Reserved.

The schematic now appears as follows.

Modifying Parameter Values for the Instrumentation Block

1. Edit the CDF parameter values for the *offset_comms_instr* as listed in Table 10-8 on page 2021.

   a. Choose *Edit – Properties – Objects* in the Schematic window.

   The Edit Object Properties form appears. You use this form to change the list of CDF (component description format) properties for *offset_comms_instr* and modify the schematic for this simulation.

   b. In the Schematic window, click the *offset_comms_instr* block.

   The Edit Object Properties form changes to display information for *offset_comms_instr* block.

   c. Change the parameter values to match those in Table 10-8.

© 1994-2020 All Rights Reserved.

d. Click OK.

Table 10-8 CDF Parameter Values for the Instrumentation Block

Parameter Name	Value
symbols per second	1228800
I-sampling delay (secs)	134n
number of symbols	2
max eye-diag volts	1
min eye volts	–1
number of hstgm bins	100
I-noise (volts^2)	0
Q-noise (volts^2)	0
statistics start time	30u
input resistance	50

Adding an Instrumentation Terminator

Add an instrumentation termination (*instr_term*) block to the right of the instrumentation block. The *instr_term* block terminates the outputs on the instrumentation block and prevents unused pin warnings.

1. In the Schematic window, choose *Add – Instance* to display the Add Instance form.

   The Add Instance form appears. It may be empty or it may display information for a previously added element.

2. In the Add Instance form, click *Browse* to display the Library Browser - Add Instance form.

3. In the Library Browser - Add Instance form, make the following selections. (If necessary, click Show Categories at the top of the Library Browser, to display the Category column.)

Library	Category	Cell	View
rfLib	*measurement*	*instr_term*	*symbol*

In the Library Browser, cell *instr_term* and it's default view *symbol* are both selected.

© 1994-2020 All Rights Reserved.

At the top of the Add Instance form,

- `rfLib` displays in the *Library*
- `instr_term` displays in the *Cell* field
- `symbol` displays in the *View* field

There are no CDF parameters for the *instr_term* cell.

4. Move the cursor over the Schematic window. The outline for the *instr_term* symbol is attached to the cursor. Move the *instr_term* block to the right of the instrumentation block and align the input pins of the instrumentation termination block with the output pins of the instrumentation block.
5. Click to place the *instr_term* block.
6. Click *Esc* to remove the symbol from the cursor.

Wiring the Termination Block to the Instrumentation Block

1. To wire the instrumentation (*offset_comms_instr)* block to the Instrumentation terminator (*instr_term)* block, in the Schematic window choose *Add - Wire (narrow)*.
2. Wire the aligned pins straight across.
3. Click *Esc* to stop wiring.

© 1994-2020 All Rights Reserved.

The schematic should look as follows.

4. In the Schematic window, choose *Design – Design Check and Save*.

   The completed schematic is verified and saved.

The schematic for the complete receiver model should look like the one in Figure 10-7 on page 2024.

© 1994-2020 All Rights Reserved.

Figure 10-7 Completed receiver model

Setting Variable Values for the Receiver Schematic

Copy the variables you entered as CDF parameters for the individual blocks from the receiver schematic to the ADE window. Then edit each variable to give it the value specified in Table 10-9 on page 2025.

1. In the ADE window, choose *Variables – Copy From Cellview* to copy the variables from the receiver schematic to the Design Variables area on the ADE window.

© 1994-2020 All Rights Reserved.

The copied variables display in the *Design Variables* area in the ADE window.

2. In the ADE window, choose *Variables – Edit* to open the Editing Design Variables form.

Adding the Values to the Copied Variables

In the Editing Design Variables form, one by one, select each variable in the *Table of Design Variables* and associate with each one, the value listed in Table 10-9.

Table 10-9 Values for Receiver Variables

Variable	Value
`lna_gain`	15

© 1994-2020 All Rights Reserved.

Table 10-9 Values for Receiver Variables

Variable	Value
`lna_ip3`	–5
`if_mx_gain`	10
`if_mx_ip`	35
`frf`	2.14G
`flo1`	2.354G
`if_rbw`	200m
`rf_rbw`	100m

To associate a value with a design variable

1. In the *Table of Design Variables*, click `lna_gain`.

   `lna_gain` displays in the *Name* field.

2. In the *Value (Expr)* field, enter the number `15`, the value from Table 10-9 on page 2025.

3. Click *Change* to list the variable name and its value from the *Table of Design Variables.*

4. Repeat these steps for the remaining variables listed in the *Table of Design Variables* to associate the values from Table 10-9 on page 2025 with the variable names.

5. Click *OK* in the Editing Design Variables form after you have added all the variable values.

© 1994-2020 All Rights Reserved.

The table of *Design Variables* in the ADE window is updated and the Editing Design Variables form is closed.

Design Variables

	Name	Value
1	flo1	2.354G
2	frf	2.14G
3	if_mx_gain	10
4	if_mx_ip	35
5	if_rbw	200m
6	lna_gain	15
7	lna_ip3	-5
8	rf_rbw	100m

Setting Up and Running a Transient Analysis

1. In the ADE window, choose *Analyses – Choose* to display the Choosing Analyses form.
2. In the Choosing Analyses form, if necessary, click *tran* to select a transient analysis.
3. In the Choosing Analyses form, enter `130u` in the *Stop Time* field.

© 1994-2020 All Rights Reserved.

4. Highlight *moderate* for *Accuracy Defaults (errpreset)*.

5. In the Choosing Analyses form, click *Options* to display the Transient Options form.
6. In the Transient Options form, enter `30u` in the *outputstart* field.

By delaying the output start, you remove start-up transients from the eye-diagrams and scatter plots.

7. Click *OK* in the Transient Options form.

© 1994-2020
All Rights Reserved.

8. Click *OK* in the Choosing Analyses form.
9. If you have not already done so, set up the simulator and model libraries with the following steps.
   a. Set the simulator options from the Simulator window as described in "Choosing Simulator Options" on page 1970.
   b. Set up the model libraries from the Simulator window as described in "Setting Up Model Libraries" on page 1972.
10. In the ADE window, choose *Simulation – Netlist and Run*.

    Messages display in the CIW. The simulation log window opens. Watch for messages stating that the simulation has completed successfully.

    Watch the CIW for messages stating the simulation is running and that it has completed successfully.

Examining the Results: Eye Diagram, Histogram, and Scatter Plot

In this section we examine the results of the transient analysis of the receiver.

Plotting the Eye Diagram (and Transient Response)

First plot an eye diagram.

1. In the ADE window, choose *Results – Direct Plot – Transient Signal*.

   The Virtuoso Schematic Editing window appears.
2. Following the prompts at the bottom of the window,

   > Select nodes or terminals, press <esc> to finish selection

   a. In the Schematic window, click the *sawtooth* net from the instrumentation (*offset_comms_instr*) block.

© 1994-2020
All Rights Reserved.

b. Click the *I_eye* net from the instrumentation (*offset_comms_instr*) block.

c. Press *Esc* to indicate that you have finished selecting outputs.

This opens a waveform window and creates a plot of the Transient Response of the net.

3. In the waveform window, double-click the X axis.

The Axis Attributes form displays:

© 1994-2020
All Rights Reserved.

a. In the *Plot vs* field, select the `sawtooth` net, such as `/net15`.

b. Click *OK*.

You see the eye-diagram shown in Figure 10-8 on page 2031.

Figure 10-8 Eye-diagram

© 1994-2020
All Rights Reserved.

The I-sampling delay parameter in the instrumentation block (*I_del*) is chosen with respect to this eye diagram. The delay is the time when the eye opens the widest.

The instrumentation block samples the input waveforms with this delay to compute all statistics and to produce scatter plots.

4. In the waveform window, choose *File – Close*.

   The waveform window closes.

Generate the Histogram

Now generate a histogram of the I-voltage at the sampling times.

1. In the ADE window, choose *Results – Direct Plot – Transient Signal*.

   The Virtuoso Schematic Editing window appears.

2. Following the prompts at the bottom of the window,

   a. Click the *eye_hist* net from the instrumentation (*offset_comms_instr)* block.

   b. Click the *eye_count_hist* net from *offset_comms_instr*.

   c. Press *Esc* to indicate that you have finished selecting outputs.

      This creates a plot in the waveform window.

3. In the waveform window, double-click the X axis.

   The Axis Attributes form displays:

   a. In the *Plot vs* field, select the *eye_hist* output, such as `/net17`.

   b. Click *OK*.

   This creates an unintelligible intermediate plot in the waveform window.

4. In the waveform window, double-click the trace in the graph.

   The Trace Attributes form displays.

   a. In the *Type/Style* cyclic field, select *Bars*.

   b. Click *OK*.

   You see a plot like Figure 10-9 on page 2033.

© 1994-2020 All Rights Reserved.

Figure 10-9 Histogram

5. In the waveform window, choose *File – Close*.

   The waveform window closes.

Generating the Scatter Plot

Generate a scatter plot of the received symbols.

1. In the ADE window, choose *Results – Direct Plot – Transient Signal*.

   This displays the waveform window.

2. Following the prompts at the bottom of the waveform window,

   In the Schematic window:

   a. Click the *I_scatter* output from the instrumentation (*offset_comms_instr)* block.

   b. Click the *Q_scatter* output from *offset_comms_instr*.

   c. Click *Esc* to indicate that you have finished selecting outputs.

      This creates a plot in the waveform window.

3. In the waveform window, double-click the X axis.

© 1994-2020 All Rights Reserved.

The Axis Attributes form displays:

a. In the *Plot vs* field, select the *I_scatter* output, such as `/net22`.

b. Click *OK*.

4. In the waveform window, double-click the trace in the graph.

The Trace Attributes form displays.

a. In the *Type/Style* cyclic field, select *Points*.

b. Click *OK*.

You see a scatter plot like Figure 10-10 on page 2034.

Figure 10-10 Scatter Plot

5. In the waveform window, choose *File – Close*.

The waveform window closes.

The Various Instrumentation Blocks

The CDMA source (*CDMA_reverse_xmit)* produced offset QPSK symbols. Offset QPSK modulation avoids traversing the origin by staggering the digital changes in the I and Q

© 1994-2020 All Rights Reserved.

signals. Running the baseband trajectory through the origin increases spectral regrowth in the transmitters.

The instrumentation block (*offset_comms_instr*) samples the I and Q signals at different times then plots the two staggered samples against each other. The resulting scatter plot shows the received symbols. A scatter plot of the unstaggered samples reveals only what is happening in one dimension, either the I or Q dimension.

For non-offset QPSK and QAM modulation schemes, use the *comms_instr* instrumentation block instead of the *offset_comms_instr* block.

Measuring RMS EVM

You can use the same instrumentation block (*offset_comms_instr*) to compute root-mean-squared error vector magnitude (RMS EVM). The error vector is the vectorial difference between the ideal received symbol and the actual received symbol.

- EVM (error vector magnitude) is the magnitude of the error vector.
- RMS EVM is the root-mean-squared value of a sequence of EVMs.

RMS EVM is one measure of a receiver's quality. RMS EVM can account for as much or as little distortion and noise as you like. The trick is to figure out where the ideal received symbol lies. You can do this using the *I_ref* and *Q_ref* inputs to the *offset_comms_instr* instrumentation block.

To calculate RMS EVM, you

- Create a duplicate copy of the receiver chain from the *BB_driver* to the *IQ_demod_BB* including these two blocks
- Place the duplicate copy below the original receiver chain in the Schematic window
- Modify the duplicate receiver chain to make it as *ideal* as you like by changing parameter values for the individual function blocks.

  For example, to see the effect of just the LNA's IP3 value on RMS EVM, in the duplicate receiver chain make the LNA's IP3 absurdly large.

Constructing the Ideal Receiver Chain

The *ideal* receiver chain (the duplicated and modified receiver chain) is driven from the same input, the *CDMA_reverse_xmit* block, as the original receiver chain. The output of the ideal receiver chain drives the instrumentation block's *I_ref* and *Q_ref* inputs.

© 1994-2020 All Rights Reserved.

In the ideal receiver chain, you copy the first receiver chain and make every block ideal.

1. Remove the *gnd* from the *I_ref* and *Q_ref* pins on the instrumentation block.

   a. In the Schematic window, choose *Edit - Delete*.

   b. In the Schematic window, click each wire and the *Gnd* symbol attached to the *I_ref* and *Q_ref* pins.

   c. Click *esc* to stop deleting.

2. Duplicate the receiver chain from the *BB_driver* to the *IQ_demod_BB* inclusive. Do not duplicate the filters. Follow the prompts at the bottom of the Schematic window.

   a. In the Schematic window, choose *Edit – Copy*.

   b. In the Schematic window, draw a box around the blocks to copy by clicking to the left of and above the *BB_driver* block and dragging the cursor to a point below and to the right of the *IQ_demod_BB* block. Click again to complete the box.

   The blocks within the box are highlighted.

   c. Click within the highlighted area.

   A copy of the highlighted blocks in the receiver chain now moves with the cursor.

© 1994-2020 All Rights Reserved.

d. Place the duplicate receiver chain so that the output pins on the *IQ_demod_BB* are in line with the *I_ref* and *Q_ref* pins on the instrumentation block.

3. Wire the duplicate receiver chain to the CDMA signal source (*CDMA_reverse_xmit)* and the instrumentation block (*offset_comms_instr*).

a. In the Schematic window, choose *Add - Wire (narrow).*

b. Connect the *IQ_demod_BB* outputs on the duplicate receiver to the *I_ref* and *Q_ref* pins on the instrumentation block.

c. Connect the output pins on *CDMA_reverse_xmit* to the input pins of the duplicate *BB_driver*. This drives the duplicate receiver from the CDMA signal source.

d. Click *esc* to stop wiring.

The schematic with the duplicate receiver chain wired up looks like Figure 10-11.

Figure 10-11 Receiver Model with Duplicated Receiver Chain

© 1994-2020 All Rights Reserved.

Modifying Parameter Values to Make the Blocks Ideal

Now modify the parameter values for each block in the duplicate receiver chain to create ideal blocks. Block names, parameter names, and parameter values are given in Table 10-10.

Table 10-10 Parameter Values to Create an Ideal Receiver

Block Names	Parameter Names	New Parameter Values
LNA_BB	*Input referred IP3 [dBm]* *{1,0,-1} for {cw, none, ccw}*	100 0
BB_butterworth_bp	*Cell Name*	BB_loss
dwn_cnvrt	*Input referred IP3 [dBm]*	100
BB_butterworth_bp	*Cell Name*	BB_loss
IQ_demod_BB	*I-[dBm] Input referred IP3* *Q-[dBm] Input referred IP3*	100 100

1. In the Schematic window, choose *Edit – Properties – Objects* to open the Edit Object Properties form.
2. In the Schematic window, select the *LNA_BB* block.

   The Edit Object Properties form changes to display properties for the *LNA_BB* block.

   a. Set *Input referred IP3 [dBm]* to 100.

   b. Set *{1, 0,-1} for {cw, none, ccw}* to 0. (This eliminates AM/PM conversion.)

   c. Click *Apply*.
3. In the Schematic window, select the first RF *BB_butterworth_bp* block.

   The Edit Object Properties form changes to display properties for the *BB_butterworth_bp* block.

   a. Change the *Cell Name* to *BB_loss*.

   b. Click *Apply*.

   The properties and symbol change to those for the *BB_loss* block. The sole purpose of the *BB_loss* model is to replace a filter in an RMS EVM analysis.

© 1994-2020 All Rights Reserved.

The *Reference impedance* for the *BB_loss* block should equal the *Output impedance* of the *BB_butterworth_bp* bandpass filter block it replaces. The value should be 50 ohms for both blocks and you should not have to change it.

The *BB_loss* model retains the filter's loss but eliminates the filter's dynamics so you can see what, if any, affect the filter has on EVM through inter-symbol interference. To eliminate the loss as well as the dynamics, you might even replace the filter with straight wires. This example uses the *BB_loss* block instead.

4. In the Schematic window, select the *dwn_cnvrt* block.

   The Edit Object Properties form changes to display properties for the *dwn_cnvrt* block

   a. Set *Input referred IP3 [dBm]* to 100.

   b. Click *Apply*.

5. In the Schematic window, select the second *BB_butterworth_bp* block.

   The Edit Object Properties form changes to display properties for the *BB_butterworth_bp* block

   a. Change the *Cell Name* to *BB_loss*.

   b. Click *Apply*.

6. In the Schematic window, select the *IQ_demod_BB* block.

   The Edit Object Properties form changes to display properties for the *IQ_demod_BB* block

   a. Set *I-[dBm] input referred IP3* to 100.

   b. Set *Q-[dBm] input referred IP3* to 100.

   c. Click *Apply*.

7. In the Edit Object Properties form, click *OK* to close the form.

8. In the Schematic window, choose *Design – Check and Save* to check and save your modifications to the circuit.

Set Up and Run a Transient Analysis

Set up and run a transient analysis as described in "Setting Up and Running a Transient Analysis" on page 2027. Set the *Stop Time* to `130u` and the *outputstart* option to `30u`. Click *OK* in both the Transient Options and Choosing Analyses forms. Choose *Simulation – Netlist and Run* to run the transient analysis.

© 1994-2020 All Rights Reserved.

Look for messages in the CIW stating that the simulation is starting. Watch the simulation log window for messages that the simulation has completed successfully.

Plot the RMS EVM Output

After the simulation, plot the RMS_EVM output of the instrumentation block.

1. In the ADE window, choose *Results – Direct Plot – Transient Signal*.

   This displays the waveform window.

2. Following the prompts at the bottom of the waveform window,

   a. In the schematic, click the *rms_EVM* output net from the instrumentation (*offset_comms_instr)* block.

   b. Click *Esc* to indicate that you have finished selecting outputs.

   This creates the RMS EVM plot in the waveform window as shown in Figure 10-12 on page 2040.

Figure 10-12 RMS EVM

© 1994-2020 All Rights Reserved.

The RMS EVM trace starts at 30us, which is the `statistics start time` parameter of the instrumentation block. The `statistics start time` parameter keeps start-up transients out of the statistics.

The trace settles out at 25.84 Volts. This means that after 130us of data is collected, and ignoring the first 30us, the RMS EVM is 25.84%. The EVM measurement is normalized to the RMS magnitude of the ideal symbol then multiplied by 100 to express the measurement as a percentage.

3. In the waveform window, choose *File – Close*.

Computing Minimized RMS Noise Using the Optimizer

There is one more construction step before proceeding to the Circuit Optimizer application. You set up the Circuit Optimizer to minimize RMS noise subject to performance constraints. This step replicates the receiver chain yet one more time to generate the noise measurement.

1. Duplicate the original receiver chain from the *BB_driver* up to and including both low pass filters (*butterworth_lp*). Follow the prompts at the bottom of the Schematic window.

   a. In the Schematic window, choose *Edit – Copy*.

   b. In the Schematic window, draw a box around the blocks to copy by clicking to the left of and above the *BB_driver* block and dragging the cursor to a point below and to the right of the *butterworth_lp* filter blocks. Click again to complete the box.

      The blocks within the box are highlighted.

   c. Click within the highlighted area.

      A copy of the highlighted blocks in the receiver chain now moves with the cursor.

© 1994-2020 All Rights Reserved.

d. Place the duplicate receiver chain above the original receiver chain.

2. In the duplicate receiver chain, ground the *Q_in* pin on the *BB_driver* block.

a. In the Schematic window, choose *Add – Wire.*

b. Click the *Q_in* pin and run the wire to the *Gnd* symbol below the *dwn_cnvrt* block.

c. Click the *Gnd* symbol below the *dwn_cnvrt* block.

d. Click *esc* to stop wiring.

3. Add a 50mV DC voltage source to the left of the *BB_driver* block to drive the *I_in* pin on the *BB_driver* block. At the same time add a *gnd* symbol below the *port* in the schematic.

a. In the Schematic window, choose *Add – Instance* to display the Add Instance form.

b. In the Add Instance form, click *Browse* to display the Library Browser - Add Instance form.

c. In the Library Browser - Add Instance form, click *analogLib*.

d. Scroll the elements in the *Cell* column and click *port*.

e. The outline for the *port* symbol is attached to the cursor. Move the *port* symbol to the left of the *BB_driver* block and click to place the *port* symbol.

f. Return to the Library Browser - Add Instance form and scroll the elements in the *Cell* column and click *gnd*.

g. The outline for the *gnd* symbol is attached to the cursor. Move the *gnd* symbol below the *port* symbol and click to place it there.

© 1994-2020 All Rights Reserved.

h. Click *Esc* to remove the *gnd* symbol from the cursor.

4. In the Schematic window, choose *Edit – Object – Properties* to modify the *port* symbol using the Edit Object Properties form.

a. In the Schematic window, click the *port* symbol.

The Edit Object Properties form changes to display information for the *port* symbol.

b. In the *Source Type* cyclic field, select *dc*.

c. In the *DC Voltage* field, enter 50m.

d. Highlight *Display small signal params* to display small signal parameters.

e. In the *AC Magnitude* field type `1V`.

f. In the *AC Phase* field type `0`.

g. Click *OK* in the Edit Object Properties form.

5. Load the low pass filters with a *res_BB* model from the *top_dwnBB* category of *rfLib*. Use the default parameters and ground the output pins.

a. In the Schematic window, choose *Add – Instance* to display the Add Instance form.

b. In the Add Instance form, click *Browse* to display the Library Browser - Add Instance form.

c. In the Library Browser - Add Instance form, click *rfLib*.

d. Scroll the elements in the *Cell* column and click *res_BB*.

The outline for the *res_BB* symbol is attached to the cursor.

e. Move the *res_BB* symbol to the right of the two low pass filters (*butterworth_lp*) and click to place the *res_BB* symbol.

f. Return to the Library Browser - Add Instance form and click *analogLib.*

g. Scroll the elements in the *Cell* column and click *gnd*.

The outline for the *gnd* symbol is attached to the cursor.

© 1994-2020 All Rights Reserved.

h. Move the *gnd* symbol to the right of the *res_BB* symbol and click to place it there.

i. Click *Esc* to remove the symbol from the cursor.

6. In the duplicate receiver chain, wire the *port*, the *res_BB* block, and their *gnd* blocks.

a. In the Schematic window, choose *Add – Wire.*

b. Click the *I_in* pin on the *BB_driver*, then click the top pin on the *port*.

c. Click the bottom pin on the *port*, then click the *gnd* pin just below it.

d. Click the *out* pin on the top *butterworth_lp* filter, then click the *I_in* pin on the *res_BB* block.

e. Click the *out* pin on the lower *butterworth_lp* filter, then click the Q_*in* pin on the *res_BB* block.

f. Click the *I_out* pin on the *res_BB* block. Then click the top pin on the *gnd* located to it's right.

g. Click the Q_*out* pin on the *res_BB* block. Then click the top pin on the same *gnd*.

h. Click *esc* to stop wiring.

7. In the Schematic window, choose *Design – Check and Save* to check and save the schematic.

The schematic with the third receiver chain is shown in Figure 10-13.

© 1994-2020 All Rights Reserved.

Figure 10-13 Schematic with Noise Generating Receiver

Set Up and Run Transient and Noise Analyses

Set up a transient analysis as described in "Setting Up and Running a Transient Analysis" on page 2027. Set the *Stop Time* to `130u` and the *outputstart* option to `30u` and make sure that the transient analysis is enabled.

Set up a noise analysis as follows:

1. In the Choosing Analyses form, click *noise* to select a noise analysis.
2. For *Sweep Variable*, click *Frequency*.
3. For *Sweep Range*, click *Start-Stop*.
4. Set up the analysis to sweep frequency from 0 to100 MHz.
   a. For the starting frequency, in the *Start* field enter `0`.
   b. For the stop frequency, in the *Stop* field enter `100M`.
5. Set up the *Output Noise* source.

© 1994-2020 All Rights Reserved.

a. In the *Output Noise* cyclic field, select *voltage.*

b. To select the *Positive Output Node*, click *Select* next to the *Positive Output Node* field. Then, in the Schematic window, click the net next to the *I_in* pin on the *res_BB* block.

c. To select the *Negative Output Node*, click *Select* next to the *Negative Output Node* field. Then, in the Schematic window, click the net next to the *I_out* pin on the *res_BB* block.

6. Set up the *Input Noise* source.

a. In the *Input Noise* cyclic field, select *port.*

b. To select the *Input Port Source*, click *Select* next to the *Input Port Source* field. Then, in the Schematic window, click the *DC input port* model.

7. Click *Enabled*.

© 1994-2020 All Rights Reserved.

The *Output Noise* and *Input Noise* sections of the Noise analysis form look as follows. (The net instance numbers might be different, however.)

8. Verify that *Enabled* is highlighted and click *OK* in the Choosing Analyses form.

9. In the ADE window, check the Analysis area to verify that both the transient and noise analyses are set up properly and that they are both enabled.

10. In the ADE window, choose *Simulation – Netlist and Run* to start the simulations.

    Watch the messages in the CIW to verify that everything is set up properly and that the simulations start. Check the simulation log window to see that the simulations run and complete properly.

© 1994-2020 All Rights Reserved.

Set Up to Run the Circuit Optimizer

1. In the ADE window, choose *Tools – Calculator* to open the calculator window.

2. In the ADE window, choose *Tools – Optimization* to open the Circuit Optimizer window.

Add the First Goal

1. In the calculator window's function panel, select *rmsNoise*.
2. When the RMS Noise panel opens,
   a. In the *From* field, enter `0`.

© 1994-2020 All Rights Reserved.

b. In the *To* field, enter `100M`.

c. Click *OK*.

The expression

`rmsNoise(0,100M)`

displays in the buffer of the calculator window.

3. In the Circuit Optimizer window, choose *Goals – Add* to add the first goal.

The Adding Goals form opens.

4. Fill in the Adding Goals form.

a. In the *Name* field, type `rmsNoise`.

b. Click the *Get Expression* button to the right of the *Calculator* label.

The expression `rmsNoise(0 100000000)` displays in the *Expression* field.

c. Ensure that the *Direction* is minimize.

The default for the Optimizer is to minimize goals.

d. In the *Target* field, enter `0.1u`.

e. In the *Acceptable* field, enter `20u`.

f. Verify that the form is enabled.

The Adding Goals form looks like this.

g. Click *OK*.

© 1994-2020 All Rights Reserved.

The first goal is added to the Circuit Optimizer window.

Add the Second Goal

1. In the calculator window, under the *tran* tab, click the *vt* button.

   The Schematic Editing window becomes active.

2. In the Schematic window, click the *rms_EVM* output net of the instrumentation block.

   An expression similar to `VT("/net24")` displays in the calculator buffer.

3. In the calculator window's function panel, select *value*.

   When the Value panel appears

   a. Enter `130u` in the *Interpolate At* field.

   b. Click *OK*.

   The expanded expression `value(VT("/net24") 130u )` displays in the buffer of the calculator window.

4. In the Circuit Optimizer, choose *Goals – Add*.

   When the Adding Goals form opens

   a. In the *Name* field, enter `evm`.

   b. Click the *Get Expression* button to the right of the *Calculator* label.

   An expression similar to

© 1994-2020 All Rights Reserved.

```
value(VT("/net24") 0.00013)
```

displays in the *Expression* field.

c. In the *Direction* cyclic field, select <=

d. In the *Target* field, enter `25`.

e. In the *Acceptable* field, enter `10`.

f. Click the *% within Target* button.

g. Verify that *Enabled* is active.

The Adding Goals form looks like this.

h. Click *OK*.

The second goal is added to the Circuit Optimizer window.

© 1994-2020 All Rights Reserved.

Add the Third Goal

1. In the calculator window, click *Clear*.

   The buffer is cleared.

2. Under the *tran* tab, click the *vt* button.

   The Schematic Editing window becomes active.

3. In the Schematic window, click the *I_in* pin of the instrumentation block.

   An expression similar to `VT("/net14")` displays in the buffer of the calculator window.

4. In the calculator window's function panel, select *rms*.

   An expanded expression similar to

   ```
   rms(VT("/net14"))
   ```

   displays in the buffer of the calculator window.

   The objective is to keep the *rms* value of this signal level above 300 mV. Note that all goals must be scalars.

5. In the Circuit Optimizer, choose *Goals - Add*.

   The Adding Goals form opens.

6. Fill in the fields of the Adding Goals form.

   a. In the *Name* field, enter `sig_level`.

   b. Click the *Get Expression* button to the right of the *Calculator* label.

   And expression similar to

   ```
   rms(VT("/net14"))
   ```

   displays in the *Expression* field.

© 1994-2020 All Rights Reserved.

c. In the *Direction* cyclic field, select >=

d. In the *Target* field, enter `300m`.

e. In the *Acceptable* field, enter `10`.

f. Click the *% within Target* button.

g. Verify that the *Enabled* button is active.

The Adding Goals form looks like this.

h. Click *OK*.

The third goal is added to the Circuit Optimizer window.

Add the Circuit Variables to the Optimizer

Add the variables to the Circuit Optimizer window.

© 1994-2020 All Rights Reserved.

1. In the Circuit Optimizer window, choose *Variables – Add/Edit*.
2. When the Editing Variables form opens
   a. In the *Name* list box, click the *lna_ip3* variable.

      The *lna_ip3* variable is highlighted in the list box and it's current value `-5` displays in the *Initial Value* field.

   b. In the *Minimum Value* field, enter `-9`.

   c. In the *Maximum Value* field, enter `10`.

   d. If necessary, click *Enabled*.

      The Editing Variables form appears as follows.

   e. Click *Apply*.

© 1994-2020 All Rights Reserved.

Information for the *lna_ip3* variable displays in the *Variables* section of the Circuit Optimizer.

3. Repeat this procedure to add all the variables and values listed in Table 10-11

Table 10-11 Variables and Values for the Optimizer

Variable Name	Initial Value	Minimum Value	Maximum Value
lna_ip3	-5	-9	10
lna_gain	15	10	30
if_mx_gain	10	1	50
if_rbw	200m	50m	300m
rf_rbw	100m	50m	300m

4. Click *OK* to close the Editing Variables form.

   In the Circuit Optimizer window, the variables and optimization goals appear as shown in Figure 10-14 on page 2056.

© 1994-2020 All Rights Reserved.

Figure 10-14 Circuit Optimizer Setup

Run the Circuit Optimizer

Note: This Circuit Optimizer analysis might take several hours to complete.

1. In the Circuit Optimizer, choose *Optimizer – Run n* to display the *Run for Fixed Number of Iterations* form.
2. In the *Number of Iterations* field, move the slider to the right until *12* displays.

3. Click *OK.*

   The *Run for Fixed Number of Iterations* form closes and the Circuit Optimizer starts running.

   Watch for simulator startup messages in the CIW. Monitor the progress of the analysis in the log window.

   The waveform window opens when the first simulation completes. As simulations complete, the results are added to the plots in the waveform window.

© 1994-2020 All Rights Reserved.

Viewing the Circuit Optimizer Output

The Circuit Optimizer results displayed in the waveform window are shown in Figure 10-15 on page 2057.

- The traces on the left show the EVM and RMS output signal level, and the RMS noise at each Circuit Optimizer iteration.
- The traces on the right show the variable function block specifications at each Circuit Optimizer iteration.

Figure 10-15 Circuit Optimizer Results After 12 Iterations

Although the example is contrived, it illustrates the use model. After the Circuit Optimizer met the constraints it tried to minimize RMS noise.

1. Save the initial state of the Analog Design Environment in case you want to start over.

© 1994-2020 All Rights Reserved.

2. Then in the Circuit Optimizer window, click *Results – Update Design*. The last click updates the variables in the Analog Design Environment window with the last set of variables found by the Circuit Optimizer. You use these states in the passband view.

Summarizing the Design Procedure

To summarize, the semi-automated design procedure consists of

1. Setting up the measurements
2. Placing tolerances on the block parameters
3. Constraining the system performance
4. Identifying a quantity to minimize (or maximize)
5. Running the Optimizer
6. Evaluating the results

This is why the process is called *semi*-automated. After evaluating results for the first or second time you probably need to

1. Refine tolerances
2. Refine goals
3. Add or delete constraints
4. Add or delete variables

Each simulation covers 100s, or about 80 CDMA symbols. The suppressed carriers are an RF carrier at 2.14 GHz, an LO carrier at 2.354GHz, and an IF carrier at 214 MHz. The symbol rate is 1.2288 Mega-symbols per second.

Creating a Passband View of the Architectural Model

After you have designed an architecture, you can quickly create a passband view of the architectural model. (Currently, the passband behavioral models in the *top_dwnPB* category and in the passband view do not introduce any specifications that are not in the baseband models.)

The passband view checks for problems that might have escaped detection in the baseband view. For example, although the baseband view quickly assesses what filters do to the baseband signal, baseband models do not indicate whether the filters are indeed removing undesired carrier harmonics.

© 1994-2020 All Rights Reserved.

Baseband modeling is also not the best way to evaluate image rejection. Although the baseband model accurately simulates how the desired signal propagates through an image rejection receiver, it does not accurately simulate how much of the image signal propagates to the receiver output.

The passband view also creates a system testbench as mentioned in "Top-Down Design of RF Systems" on page 1962.

Procedures for Creating the Passband Model of the Receiver

The procedures described in this section illustrate how to

- Switch from a baseband to a passband view
- Make an end-to-end RF measurement
- Measure the one dB compression point

  The one dB compression point is usually a transmitter specification but it is used to demonstrate this flow because it is easier to set up.

Perform the following steps to create the passband model of the receiver:

1. Copy the original receiver model from the Circuit Optimizer analysis to a new schematic window. Copy everything from the LNA to the low pass filters.
2. Edit the properties of the IQ_demodulator (`IQ_demod_BB`) to set the last parameter, *flo*, to *-frf+flo1*. The baseband view does not need the local oscillator frequency but the passband view does.
3. Load the low pass filters with ports.
4. Connect a port across the LNA_BB inputs. Set the Frequency name to `fin`, the frequency to `frf`, and the amplitude to `power`. (Do not abbreviate `power` to `pwr` because `pwr` is a reserved variable and you do not get any warning. SpectreRF may complain about a mysterious indexed undefined variable that increments from run to run.)
5. Add loaded voltage-controlled-voltage sources as shown in Figure 10-16 on page 2060 to observe intermediate differential voltages.

© 1994-2020 All Rights Reserved.

Figure 10-16 Passband View

6. Check and save the schematic
7. Close the schematic window.
8. Bring up a Library Manager and select but do not open the schematic you just created.
9. In the Library Manager window click *File – New – Cell View*.
10. In the window that comes up, select Hierarchy Editor for the Tool. Click *OK*.
11. Type in `schematic` for the View then click *Use Template*.
12. Set the *Name* to `Spectre`.
13. Enter `veriloga_PB` as the first item in the View List, then click *OK*.
14. In the Hierarchy Editor window, click *File – Save*. This is important.
15. Click the *Open* button to bring up the schematic.
16. Bring up an Analog Design Environment tool and use the states from the last Circuit Optimizer iteration.
17. Recall the states you saved from the last Circuit Optimizer iteration. Add the `power` variable and set it to -16.
18. Delete the previous analyses and set up a PSS analysis. Enter a new Fundamental Tone named `LO` and make it 2.354GHz. *Auto Calculate* the *Beat Frequency* and type `1` for the *Number of harmonics*.
19. Run the analysis and observe the intermediate differential voltages. The model is indeed now a passband model. At the higher power levels the LNA output contains odd

© 1994-2020 All Rights Reserved.

harmonics of the RF carrier. The filter reflects the odd harmonics back to the LNA and does not let them propagate forward. The baseband model does not simulate the odd harmonics but it does simulate the intermodulation term between the second harmonic and fundamental that falls at the fundamental. One reason to simulate the passband view is to check for peak voltage levels that might exceed voltage ratings. The baseband models only simulate peak voltage at the carrier fundamental, not the absolute peak.

20. Set up a swept PSS analysis. Sweep *power* from -32 to 0 in 10 steps.

21. After the sweep finishes, click *Results – Direct Plot* in the Environment window, select *Compression point*, *1dB*, *Output Referred*. Select the 0 harmonic because the end-to-end system produces a baseband output. Then click the port loading the top output low pass filter. You should see the compression point plot shown in Figure 10-17 on page 2061.

Figure 10-17 End-to-end RF measurement, one dB compression point at the I-baseband output.

22. Repeat the last step but this time click the lower output port. You should see the compression plot in Figure 10-18 on page 2062.

© 1994-2020 All Rights Reserved.

Figure 10-18 End-to-end RF measurement, one dB compression point at the Q-baseband output.

Comparing Baseband and Passband Models

This section illustrates how to compare baseband and passband models by:

1. Setting up a Transient analysis with the passband view
2. Setting up a Transient analysis with the baseband view
3. Directly comparing the baseband and passband models.

You run one analysis of the baseband view and two analyses of the passband view. You perform the second passband analysis with tightened tolerances.

1. Save the passband schematic under a different name. You use the new copy.
2. Repeat steps 9 through 17 from the last recipe for the new copy but do not enter the `veriloga_PB` view in the *View List* yet. You do a baseband analysis first.

© 1994-2020 All Rights Reserved.

3. Delete the port driving the LNA.
4. Delete the loaded voltage-controlled-voltage sources.
5. You need to synthesize an antenna signal. Add an `IQ_mod_BB` from the `top_dwnBB` category. Set the I and Q gains to 0 dB. Set the 1dB compression points to 1000 so that the modulator is ideal. Instantiate it in front of the LNA with the pins aligned, then wire the pins straight across. Ground the `phase_err` pin.
6. Drive the `I_in` pin of the IQ modulator with a port. Set the port frequency to 2MHz and name the frequency BB1. Set the amplitude to -16dBm.
7. Do the same for the `Q_in` modulator input.
8. Load or duplicate the states from the 12-iteration Circuit Optimizer analysis but delete the Noise and Transient analyses.
9. Remove AM/PM conversion from the LNA by setting the last parameter in the properties list to zero.

   It is not fair to compare passband and baseband views with AM/PM conversion because the passband view does not capture it.
10. Set up a 1us Transient analysis with default options.
11. Run the analysis and plot the filtered baseband outputs, the outputs of the low pass filters.

    Note how fast the simulation runs. Save the results so you can plot them again later. [1]
12. Switch to the passband view by entering “veriloga_PB” in the View List in the Hierarchy Editor. Click the update button in the hierarchy editor.
13. After you switch to the `veriloga_PB` view, edit the IQ modulator properties to set *flo* to `frf`. Edit the demodulator properties and set its *flo* to `flo1-frf`.
14. Click *Results – Printing Plotting Options* then click the *Overlay Plots* button.
15. Overlay the passband results with the baseband results. You see the waveforms in Figure 10-19 on page 2064. The comparison is not very good.
16. Rerun the analysis with conservative options and set reltol to 1e-6. This run takes longer.
17. Plot the results.

---

1. Note that the baseband outputs are out of phase with each other, even though the baseband inputs are in phase. In the baseband model, changing the RF-IF mixer LO from “flo1” to “-flo1” fixes the sign problem. In the passband model, the IQ_demodulator flo should be frf-flo1. To maintain the convention, in the baseband model the IF filter’s carrier frequency should be frf-flo1.

© 1994-2020 All Rights Reserved.

18. Recall the saved baseband results and overlay them with those from the last simulation. You see the waveforms in Figure 10-20 on page 2065. The passband results now lay right on top of the baseband results but took much longer to compute! It was not obvious without the baseband results that the first passband simulation did not run with tight enough numerical tolerances.

Figure 10-19 Passband and baseband results with default options in the passband analysis

© 1994-2020
All Rights Reserved.

Figure 10-20 Passband and baseband results with tighter options in the passband analysis

Relationship Between Baseband and Passband Noise

Noise analysis at baseband can be confusing because factors of two appear in a number of places throughout the calculations. For example:

- Each passband node becomes two equivalent baseband nodes.
  - Does noise injected at a passband node split between the two equivalent baseband nodes?
  - If so, does the noise split evenly?
- As shown in the `BB_test_bench` example, baseband models simulate peak in-phase and peak quadrature components of the carrier.

  When analyzing signal-to-noise ratios, does that mean you have to use half the square of the baseband signals?
- The analog design environment displays single sided power spectral densities.

© 1994-2020 All Rights Reserved.

Because the baseband power spectral density is the two-sided passband density shifted down, is there another factor of two because we can only see the baseband density for positive frequencies?

- White noise at the input of a mixer mixes up to the carrier from DC but there is also noise at twice the carrier that mixers down to the carrier.

  Does the baseband model account for this?

These questions are answered in this section.

Before sorting out the factors of two, note that baseband noise analysis is valid only for small signals. If any element in the architecture operates in a non-linear fashion, the noise analysis might be inaccurate. This is due to the fact that a baseband noise analysis follows a DC operating point analysis, rather than a PSS analysis.

Instantaneous incremental gain in a passband static non-linear model dithers at the carrier frequency.

- When the carrier swings through zero, the incremental gain is large and noise at the input is amplified.
- When the carrier reaches its peak and drives the circuit into saturation, the incremental gain is smaller.

The average gain is greater than the minimum gain. The baseband model remains in the non-linear region because it only simulates peak voltages. Consequently, the incremental gain is always a minimum and the baseband model under-estimates the amount of noise propagating to the output. If the peak input signal drives the model into saturation, be sure to scale the baseband noise results accordingly.

Introduction to Analysis

The circuit discussed here is called `noise_test_circuit` and you can find it in the `rfExamples` library. The circuit looks as shown in Figure 10-21 on page 2067.

© 1994-2020 All Rights Reserved.

Figure 10-21 The noise_test_circuit in the rfExamples Library

The `noise_test_circuit` shows the relationship between baseband and passband noise. One branch consists of passband models. The other branch is a baseband equivalent of the first. You can assess noise at each of three observation points located in each branch of the circuit. At each observation point, you can examine both the noise and pnoise summaries.

The I and Q inputs are both driven by the same DC source so that you only have to view one baseband output, the other baseband output is identical by symmetry. Noise parameters in the passband and baseband models are identical. Aside from the behavioral blocks at the end of each branch, each behavioral block has noise injected at its input.

Preparation Steps for Analyses

1. Set up a PSS analysis.

   Because the local oscillator is inside the passband mixer models you have to manually enter the frequency (1GHz) into the PSS analysis form. Let the beat frequency be *autocalculated* and use 1 harmonic.

2. Set up a pnoise analysis.

   Set the start frequency equal to 0 Hz, the stop frequency equal to 100MHz, use a linear sweep with 100 steps. Set the Maximum sideband to 1. For the input source select *none*

© 1994-2020 All Rights Reserved.

and for the output use *voltage*. For the positive output node, select the I-input of the IQ_modulator in the passband branch. Use ground for the negative node.

3. Set up a noise analysis.

   Sweep the frequency from 0 to 100MHz linearly in 100 steps. Use voltage for the output noise and select the I-input of the *IQ_mod_BB* component in the baseband branch for the positive node. Select ground for the negative node. Set the input noise port to any one of the ports in the circuit. Noise from that port does not affect either passband or baseband branches.

4. Run the analysis.

5. When the analysis finishes, go to the analog design environment simulation window and click *results – print – pss noise summary*.

6. In the form that appears, include All types, set *Type* to *integrated noise*, look at noise from 0 to 100MHz, select *truncate by number*, and view the top 2 noise contributors.

   There are only 2 noise contributors at this point, noise at the I/Q inputs and noise from the low pass resistors.

7. Print the noise summary (which is different from the pnoise summary).

   Again, print integrated noise (from 0 to 100MHz). Select All types and print noise from the top 2 noise contributors.

   Figure 10-22 on page 2069 shows the noise summaries. The two summaries agree because at this point in the circuit, both nodes are really baseband nodes.

© 1994-2020 All Rights Reserved.

Figure 10-22 Noise Summaries for the First PSS pnoise and Noise Analyses

```
Device    Param             Noise Contribution    % Of Total

/I2       IQ_modulator_i    1.56676e-12           70.06
/R20      rn                6.69678e-13           29.94

Integrated Noise Summary (in V^2) Sorted By Noise Contributors
Total Output Noise = 2.23644e-12
No input referred noise available


Device    Param          Noise Contribution    % Of Total
/I22      IQ_mod_BB_i    1.56676e-12           70.06
/R22      rn             6.69679e-13           29.94

Integrated Noise Summary (in V^2) Sorted By Noise Contributors
Total Output Noise = 2.23644e-12
Total Input Referred Noise = Inf
```

8. Repeat the analyses.

This time select the `outi` pins on the low pass filters as outputs in the appropriate noise analyses. Again print the same noise summaries but this time look at the top 12 noise contributors in each summary.

Figure 10-23 on page 2070 shows the noise summaries for the second analysis. The top 7 noise contributors in the baseband and passband branches agree. The remaining noise contributors are negligible, and should be negligible for the circuit because it has no AM/PM conversion.

© 1994-2020 All Rights Reserved.

Figure 10-23 Noise Summaries for the Second PSS pnoise Analyses

```
Device      Param                     Noise Contribution    % Of Total

/I77        PA_PB                     7.49879e-10           82.29
/I2         IQ_modulator_i            9.69448e-11           10.64
/R20        rn                        4.14371e-11           4.55
/I87        butterworth_lp_i_noise    1.52928e-11           1.68
/I94        IQ_demodulator            7.64642e-12           0.84
/PORT13     rn                        3.77571e-14           0.00
/PORT14     rn                        3.77571e-14           0.00
/I2         IQ_modulator_q            2.33139e-15           0.00
/R21        rn                        9.96506e-16           0.00
/R22        rn                        0                     0.00
/R23        fn                        0                     0.00
/R23        rn                        0                     0.00

Integrated Noise Summary (in V^2) Sorted By Noise Contributors
Total Output Noise = 9.11279e-10
No input referred noise available


Device      Param                     Noise Contribution    % Of Total
/I76        PA_HB_i                   7.49861e-10           82.29
/I22        IQ_mod_BB_i               9.69448e-11           10.64
/R22        rn                        4.14371e-11           4.55
/I86        butterworth_lp_i_noise    1.52928e-11           1.68
/I92        IQ_demod_HB_i             7.64642e-12           0.84
/PORT15     rn                        3.77571e-14           0.00
/PORT9      rn                        3.77571e-14           0.00
/I76        PA_HB_q                   1.80329e-14           0.00
/I22        IQ_mod_BB_q               2.33136e-15           0.00
/R23        rn                        9.96493e-16           0.00
/I82        IQ_demod_HB_q             1.54184e-31           0.00
/R23        fn                        0                     0.00

Integrated Noise Summary (in V^2) Sorted By Noise Contributors
Total Output Noise = 9.11279e-10
Total Input Referred Noise = 0.0151541
```

9. Repeat the analysis again.

This time change the Pnoise sweep to run from 900MHz to 1.1GHz in 200 linear steps and select the power amplifier output as the noise output node. For the noise analysis, leave the sweep at zero to 100MHz, use 100 steps as before, and change the output node to be the I-output of the power amplifier.

© 1994-2020 All Rights Reserved.

Look at the top 7 noise contributors in each analysis. This time, integrate noise from 900MHz to 1.1GHz for the pnoise run and integrate noise from 0 to 100MHz for the noise run.

Figure 10-24 on page 2071 shows the new summaries. Although the noise analyses agree at the ends of the branches, the noise analyses appear to disagree at a point where the baseband node is only a baseband equivalent, not a true baseband node.

Figure 10-24 Noise Summaries for the Third PSS pnoise Analyses

```
Device      Param              Noise Contribution    % Of Total

/I77        PA_PB              2.3769e-09            83.70
/I2         IQ_modulator_i     1.53663e-10           5.41
/I2         IQ_modulator_q     1.53663e-10           5.41
/R20        rn                 6.568e-11             2.31
/R21        rn                 6.568e-11             2.31
/I84        IQ_demodulator     2.42376e-11           0.85
/PORT13     rn                 1.19682e-13           0.00

Integrated Noise Summary (in V^2) Sorted By Noise Contributors
Total Output Noise = 2.83995e-09
No input referred noise available


Device     Param              Noise Contribution    % Of Total
/I76       PA_BB_i            2.3769e-09            83.69
/I22       IQ_mod_BB_i        3.07294e-10           10.82
/R22       rn                 1.31347e-10           4.62
/I82       IQ_demod_BB_i      2.42376e-11           0.85
/PORT9     rn                 1.19682e-13           0.00
/I76       PA_BB_q            5.71606e-14           0.00
/I22       IQ_mod_BB_q        7.38993e-15           0.00

Integrated Noise Summary (in V^2) Sorted By Noise Contributors
Total Output Noise = 2.83997e-09
Total Input Referred Noise = 0.0148587
```


© 1994-2020 All Rights Reserved.

The *apparent* disagreement shown in Figure 10-24 on page 2071 requires an explanation. Let us examine the noise contributors and try to answer some the questions we posed earlier.

Noise at the power amplifier output due to noise injected at a passband node

- Passband model contributors: *PA_PB* and *IQ_demodulator*, port 9.
- Baseband model contributors: *PA_BB_i*, *IQ_demod_BB_i*, port 13.

Passband and baseband counterparts contribute the same amount of noise. However, in the baseband model, from symmetry you see the same numbers if you look at the Q-node. This means the baseband model predicts twice as much total noise due to noise injected between the modulators.

This factor of two is intentionally introduced to maintain the correct signal-to-noise ratio. The baseband model simulates peak signals; the carrier is suppressed. Without the carrier, signal power equals the square of the peak rather than one half of the square. This factor of two is not as arbitrary as it seems. The baseband model predicts the correct noise after demodulation because the passband demodulator model includes an extra factor of two to offset the factor of two inherent in the demodulation process.

Let the modulated carrier be `i(t)*cos(wc*t) -q(t)*sin(wc*t) + noise(t)`.

Now consider the I-output. To generate the I-output, the demodulator multiplies the signal by `cos(wc*t)`. The only part that propagates through the subsequent filter is `(1/2)i(t) + noise(t)*cos(wc*t)`. To recover `i(t)`, the passband demodulator model must scale this sum by two. (The baseband demodulator does not need to scale by two to extract the baseband signal because the carrier is suppressed.)

Thus, noise at passband demodulator model output equals `2*noise(t)*cos(wc*t)`. The filtered noise power density is then `4*(`*`input noise density`*`)/2`. The factor of 1/2 comes from the cosine. The filtered output noise density is twice the input noise density.

In the baseband model, doubling the noise injected at the passband nodes was not simply a matter of convenience.

So, to answer questions 1 and 2,

- Yes, noise injected at a passband node splits evenly between the two equivalent baseband nodes

  but

© 1994-2020 All Rights Reserved.

- Because each split is doubled, the ratio of *peak* signal to total noise equals the true signal-to-noise ratio.

Question 3 is rendered moot by using the noise summary and integrating over the proper band. If the pnoise analysis only integrated from 1GHz to 1.1GHz, (instead of from 900MHz to 1.1GHz), there would be a mysterious factor of two error.

In the baseband model, phase noise entering on the phase error pin propagates to both the I and Q outputs. In the baseband model, the same noise power appears on just the one output. Again, the total noise in the baseband model is twice that of the passband model to maintain the correct signal-to-noise ratio.

Noise injected at the modulator input (resistors and modulator noise)

Total noise in the baseband model due to modulator and input resistor noise is twice what it is from just one phase. Thus, the total noise due to sources on the input sides of the modulators differ by a factor of two. This occurs because the passband model is a real multiplier, which modulates the noise. If the peak signal voltage agrees with the baseband model, the passband modulator model attenuates input noise most of the time. The important thing is that the signal-to-noise ratios in the passband and baseband models agree anywhere in the system.

Now, copy the circuit, remove the capacitors at the modulator inputs, and repeat the last set of noise analyses. You see that in the passband model, the input resistors, R20 and R21, together contribute twice as much as the baseband counterpart, R22. With the capacitors, R20 and R21 together contribute just as much as R22. Without the capacitors, the input noise is truly white over the frequencies of interest. The same thing happens to the modulator noise itself. Figure 10-25 on page 2074 shows the results.

In particular, the modulator now also has noise at twice the carrier frequency and that noise mixes down to the carrier frequency. The baseband model is just that, a baseband model. The answer to question 4 is *no*. The baseband models do not account for noise, or signals, at carrier harmonics. The baseband equivalent noise analysis is valid only if noise injected into the modulators has no power beyond the local oscillator frequency. Phase noise injected at the phase noise pins should also be band limited.

© 1994-2020 All Rights Reserved.

Figure 10-25 Noise Summaries with the input capacitors removed.

Device	Param	Noise Contribution	% Of Total
/I2	IQ_modulator_q	4.22571e-09	29.21
/I2	IQ_modulator_i	4.22571e-09	29.21
/I77	PA_PB	2.3769e-09	16.43
/R21	rn	1.80619e-09	12.49
/R20	rn	1.80619e-09	12.49
/I84	IQ_demodulator	2.42376e-11	0.17

Integrated Noise Summary (in V^2) Sorted By Noise Contributors
Total Output Noise = 1.44651e-08
No input referred noise available

Device	Param	Noise Contribution	% Of Total
/I22	IQ_mod_BB_i	4.22561e-09	50.11
/I76	PA_BB_i	2.3769e-09	28.19
/R22	rn	1.80615e-09	21.42
/I82	IQ_demod_BB_i	2.42376e-11	0.29
/PORT9	rn	1.19682e-13	0.00
/I22	IQ_mod_BB_q	1.01619e-13	0.00

Integrated Noise Summary (in V^2) Sorted By Noise Contributors
Total Output Noise = 8.43322e-09
Total Input Referred Noise = 9.76997e-11

© 1994-2020 All Rights Reserved.

11

Documents That Ship in the Software Hierarchy

In addition to the SpectreRF documents visible through CDNSHelp, there are many documents that ship in the software hierarchy along with workshops. To find these documents, go to

`$SPECTREHOME/tools.lnx86/spectre/examples/SpectreRF_workshop`

The structure of the `SpectreRF_workshop` directory is given in the following table. To make the best use of this information, look at the summaries of contents to determine which document contains the information you need. That document will tell you which tar ball contains the corresponding database. The tar balls are also included in the `SpectreRF_workshop` directory, but are not listed here.

Document or directory	For a brief summary of contents, see
`SpectreRF_simulink_example.pdf`	SpectreRF_simulink_example.pdf on page 2076
`EnvelopeAN.pdf`	EnvelopeAN.pdf on page 2077
`HB_AN.pdf`	HB_AN.pdf on page 2079
`JitterAN.pdf`	JitterAN.pdf on page 2082
`LSSP_AN.pdf`	LSSP_AN.pdf on page 2077
`LTJM_AN.pdf`	LTJM_AN.pdf on page 2080
`MatlabAN.pdf`	MatlabAN.pdf on page 2082
`MatlabWorkshop.pdf`	MatlabWorkshop.pdf on page 2077
`NS_AN.pdf`	NS_AN.pdf on page 2078
`PerturbationAN.pdf`	PerturbationAN.pdf on page 2080
`PLL_Jitter_AN.pdf`	PLL_Jitter_AN.pdf on page 2078

© 1994-2020 All Rights Reserved.

Document or directory	For a brief summary of contents, see
PSRR_Drv_AN.pdf	PSRR_Drv_AN.pdf on page 2079
PSRR_Osc_AN.pdf	PSRR_Osc_AN.pdf on page 2081
PstbAN.pdf	PstbAN.pdf on page 2083
readme.txt	readme.txt on page 2079
RF_Blocks_AN.pdf	RF_Blocks_AN.pdf on page 2081
hbsp.pdf	
OFDM_tutorial.pdf	
GFSK_tutorial.pdf	

SpectreRF_simulink_example.pdf

Title: *Co-Simulation with SpectreRF MATLAB/Simulink Tutorial*

This document is a tutorial that describes how to run a co-simulation between SpectreRF and MATLAB/Simulink.

Tutorial 1 demonstrates Simulink, SpectreRF co-simulation while the SpectreRF simulator runs a tran analysis, for the system-level module of a wireless LAN. The three steps required for the co-simulation include:

1. Inserting and configuring the SpectreRF Engine in the Simulink schematic.
2. Setting up the netlist to adopt the SpectreRF and MATLAB co-simulation.
3. Running the co-simulation.

Tutorial 2 demonstrates Simulink, SpectreRF co-simulation while the SpectreRF simulator runs an ENVLP analysis. This tutorial illustrates:

1. The use mode of manually running Spectre with ADE and calling MATLAB from ADE.
2. The use mode of manually running Simulink and activating Spectre through Simulink.

Note: You can download the PDF from the `$SPECTREHOME/tools.lnx86/spectre/examples/SpectreRF_workshop` directory.

© 1994-2020 All Rights Reserved.

EnvelopeAN.pdf

Title: *A User's Guide to Envelope Following Analysis Using SpectreRF*

This document is an application note that describes how to use envelope following analysis with SpectreRF. The topics covered include:

1. Envelope following algorithm
2. Using the envelope following analysis
3. Parameters used in the analysis
4. Plotting results
5. An automatic gain control (AGC) example
6. Comparing simulation and measurement results for a GSM power amplifier
7. Using autonomous envelope following, with examples
8. Simulating circuits with FM sources

Note: You can download the PDF from the `$SPECTREHOME/tools.lnx86/spectre/examples/SpectreRF_workshop` directory.

LSSP_AN.pdf

Title: *Measuring Large-Signal S-Parameters Using SpectreRF*

This document illustrates how to measure large-signal S-parameters using Spectre and SpectreRF from within ADE. The contents include:

1. Calculating LSSP in terms of input power
2. Calculating LSSP in terms of beat frequency
3. Comparing large and small-signal S-parameters

Note: You can download the PDF from the `$SPECTREHOME/tools.lnx86/spectre/examples/SpectreRF_workshop` directory.

MatlabWorkshop.pdf

Title: *SpectreRF Workshop: Using the SpectreRF MATLAB Toolbox*

© 1994-2020 All Rights Reserved.

This document is a workshop that describes how to use SpectreRF and the SpectreRF MATLAB toolbox to measure parameters that are important in the design verification of low noise amplifiers (LNA). The contents include:

1. Measuring the small signal gain (SP) of an LNA
2. Conducting a large signal noise analysis (PSS and Pnoise)
3. Measuring gain compression and total harmonic distortion (Swept PSS)
4. Measuring IP3 (PSS plus PAC)
5. Measuring IP3 (QPSS analysis with Shooting or the Harmonic Balance engine)
6. Measuring IP3 (Rapid IP3 using AC analysis)
7. Measuring IP3 (Rapid IP3 using PSS Plus PAC)

Note: You can download the PDF from the `$SPECTREHOME/tools.lnx86/spectre/examples/SpectreRF_workshop` directory.

PLL_Jitter_AN.pdf

Title: *Application Note. Pll jitter measurements*

This document illustrates how to use the Spectre and SpectreRF simulators with ADE to measure jitter characteristics of phase-locked loop (PLL) circuits. The contents include:

1. Using SpectreRF to simulate the individual blocks and measure the jitter and operating parameters of the models
2. Creating or modifying the behavioral models of the blocks to incorporate the jitter
3. Time domain simulation of the PLL using the behavioral models of the blocks.
4. Post processing of the simulation results to obtain the jitter and noise characteristics of the entire PLL.

Note: You can download the PDF from the `$SPECTREHOME/tools.lnx86/spectre/examples/SpectreRF_workshop` directory.

NS_AN.pdf

Title: *The Application Note of Noise Separation in Pnoise/Qpnoise Analysis*

© 1994-2020 All Rights Reserved.

This document is an application note that describes how to use noise separation features in Pnoise and Qpnoise analysis to extract noise sources. The contents include:

1. The principles of noise separation in RF circuits
2. The graphical user interface used for noise separation
3. The flow to follow to determine which noise sources add the most noise to the output
4. Understanding the noise source summary report
5. An example is provided that illustrates how to locate the noise sources that contribute the most noise to the output in an NE600 mixer.

Note: You can download the PDF from the `$SPECTREHOME/tools.lnx86/spectre/examples/SpectreRF_workshop` directory.

PSRR_Drv_AN.pdf

Title: *Power Supply Rejection Ratio Characterization Using SpectreRF Driven Circuits*

This document illustrates how to characterize the power supply rejection ratio (PSRR) of driven circuits. The contents include:

1. How to set up the PXF analysis
2. Using Direct Plot for the PXF analysis to plot the periodic transfer function

Note: You can download the PDF from the `$SPECTREHOME/tools.lnx86/spectre/examples/SpectreRF_workshop` directory.

readme.txt

This text document lists the main topic of each of the `.pdf` files in the `SpectreRF_workshop`.

HB_AN.pdf

Title: *Harmonic Balance Release Note*

© 1994-2020 All Rights Reserved.

This document illustrates how to use the Harmonic Balance (HB) method, which is very efficient in simulating circuits with only low order harmonics, such as low-noise amplifiers. The contents include:

1. The parameters used to specify HB for driven circuit large-signal simulations
2. How to use HB from within ADE
3. Discussion of when it is most appropriate to use time domain Shooting and when it is better to use HB

Note: You can download the PDF from the `$SPECTREHOME/tools.lnx86/spectre/examples/SpectreRF_workshop` directory.

LTJM_AN.pdf

Title: *Application Note. Long term jitter measurements*

This document describes the procedures used to measure long term jitter characteristics from within the Analog Design Environment (ADE). Long term jitter characterizes the variation in the accumulated width of a large number of clock periods. The contents include:

1. Plotting the phase noise power spectral density, which will be used to compute the long term jitter
2. Selecting the number of cycles to use for the measurement
3. Selecting the minimum frequency of the integration. Long term jitter is described as very sensitive to the lower frequency limit that is used in the integration of the phase noise.

Note: You can download the PDF from the `$SPECTREHOME/tools.lnx86/spectre/examples/SpectreRF_workshop` directory.

PerturbationAN.pdf

Title: *Perturbation Based Measurements Using SpectreRF*

This document describes how to use perturbation based rapid IP3, rapid IP2, the IM2 distortion summary, and the compression distortion summary to measure the compression distortion and intercept points of RF circuits. Contents include:

1. A discussion of intermodulation distortion
2. A discussion of compression distortion

© 1994-2020 All Rights Reserved.

3. A discussion of the perturbation method used in the analyses
4. An example using IP3 and compression distortion summary measurements of a mixer
5. An example using IP2 and IM2 distortion summary measurements of a Gilbert direct conversion mixer
6. An example using IPa3 and compression distortion summary measurements of a power amplifier

Note: You can download the PDF from the `$SPECTREHOME/tools.lnx86/spectre/examples/SpectreRF_workshop` directory.

PSRR_Osc_AN.pdf

Title: *Power Supply Rejection Ratio Characterization Using SpectreRF Autonomous Circuits*

This document describes how to measure the power supply rejection ratio (PSRR) of oscillators and other autonomous circuits. Knowing PSRR values can help you protect against noise sources such as power supply ripples. The same methodology can be used to measure the effect of substrate noise. Contents include:

1. Setting up the PXF analysis to measure the PSRR of a VCO.
2. A discussion of the different definitions of PSRR
3. A discussion of confirming the results of the PXF analysis by taking quasi-static measurements of the tuning sensitivity and frequency pushing of the VCO

Note: You can download the PDF from the `$SPECTREHOME/tools.lnx86/spectre/examples/SpectreRF_workshop` directory.

RF_Blocks_AN.pdf

Title: *Using Verilog-A Baseband Models of RF Blocks Application Note*

This document describes how to create and use Verilog-A signal flow and conservative baseband models of RF blocks. Contents include:

1. Several examples exploring the correspondence between passband (PB) and baseband (BB) representations of RF signals
2. Examples illustrating baseband modeling of linear time invariant blocks

© 1994-2020 All Rights Reserved.

3. Examples illustrating baseband modeling of nonlinear blocks using an amplifier
4. Code examples for Verilog-A modules, including
   - An ideal baseband to passband converter
   - A behavioral model of a passband to baseband converter
   - A behavioral model of a deviation frequency meter
   - A baseband model of a passband low pass filter with transfer function `H(s) = a/(a+s)`
   - A signal flow baseband model of a nonlinear RF block based on the complex BB gain approach

Note: You can download the PDF from the `$SPECTREHOME/tools.lnx86/spectre/examples/SpectreRF_workshop` directory.

JitterAN.pdf

Title: *Jitter Measurements Using SpectreRF Application Note*

This document illustrates how to measure the jitter characteristics of typical blocks, working within ADE. Examples are presented for both driven and autonomous circuits. Contents include:

1. A discussion of how time (or phase) jitter is defined. RMS jitter and peak-to-peak jitter measurements are introduced
2. An example illustrating how to measure jitter for an autonomous circuit (differential VCO)
3. An example illustrating how to measure phase noise and jitter at the output of a divider, in a circuit that uses a VCO with divide-by-2 prescaler

Note: You can download the PDF from the `$SPECTREHOME/tools.lnx86/spectre/examples/SpectreRF_workshop` directory.

MatlabAN.pdf

Title: *Spectre/RF Matlab Toolbox Application Note*

This document describes how to use SpectreRF MATLAB toolbox to read in a simulation result and perform standard RF measurements. Contents include:

1. Instructions for obtaining and setting up the toolbox

© 1994-2020 All Rights Reserved.

2. A list and description of the basic functions included in the toolbox:
   - `cds_srr`
   - `cds_evalsig`
   - `cds_plotsig`
   - `cds_harmonic`
   - `cds_interpsig`
   - `cds_fft`
   - `cds_compression`
   - `cds_ipn`
3. An example illustrating how to obtain the 1dB compression point, the first order harmonic signal, and the total harmonic distortion
4. A discussion of compatibility with the Aptivia MATLAB functions

Note: You can download the PDF from the `$SPECTREHOME/tools.lnx86/spectre/examples/SpectreRF_workshop` directory.

PstbAN.pdf

Title: *Stability analysis of Linear Periodical Time-Varying Circuit using SpectreRF PSTB Analysis Application Note*

This document discusses how to use STB analysis to evaluate the stability of a linear periodical time-varying circuit. Contents include:

1. An example illustrating how to run an STB and PSTB analysis on a VCO with an inherent nonlinearity
2. An example illustrating how to examine the local stability of an injection-locked oscillator

An example illustrating how to examine the global stability of an injection-locked oscillator

Note: You can download the PDF from the `$SPECTREHOME/tools.lnx86/spectre/examples/SpectreRF_workshop` directory.

© 1994-2020 All Rights Reserved.

© 1994-2020 All Rights Reserved.

Index

A

Accuracy and Settings Trade-Offs 687
additionalParams 748, 886
Aliasing 89
analysis
  descriptions 647, 840, 1117
annotate 153, 323, 398, 725
Axis Cut 85

B

beat frequency 678
before Simulation value of Start MATLAB field 1951
Bit Waveform Examples 1800
Bit Waveform Parameters 1797

C

capq 1918
checkpss 737
cmin 151, 708
Commonly Used HB Options 147
Commonly Used hbac options 319
compression 154, 730
Compression Analysis 126
Compression Distortion Summary 352
Compression Point 195, 250, 785
Compression Vs Output Power 794
Controlling Accuracy 672
Controlling Rational Fit Accuracy 1889
conventions, typographic and syntax 23
Convergence 101, 662
Convolution-Based Method 1869
cosimulation
  generating netlist for lower-level block 1948
  inputs, selecting 1949
  options, enabling 1950
  outputs, selecting 1949
  running 1957
  setting up and running 1942
  simulation response timeout, setting 1945
  socket mode for, setting 1945
  socket port 1950
  socket port for, setting 1945
  software required for 1942
  Spectre command used in 1945
  start method for 1947, 1951
  starting ADE manually and MATLAB automatically 1952
  starting MATLAB and Spectre RF separately 1952, 1957
  starting MATLAB manually and Spectre RF automatically 1954, 1958
  starting Spectre RF manually and MATLAB automatically 1958
  stop time 1952
  with MATLAB, introduction to 1942
Cosimulation with MATLAB 1942
coupler block, connecting to system-level Simulink schematic 1943

D

dBm 759
DC Principles 660
Dcblock 1906
Dcfeed 1902
Delay time 677, 1733
delayline element 1713
Designer's Guide to SPICE and Spectre 22
Diamond Cut 122
Diamond Cut with Three Frequencies 82
Distortion Summary 306
documents, shipped in the software hierarchy 2075
Down conversion 288, 841

E

Envelope Analysis 30
envlp analysis 1587
errpreset 101, 679
Exponential Waveform Parameters 1795

© 1994-2020 All Rights Reserved.

F

Fall time 1733
fdharms 724
finitediff 716
Format — Port/Signal Displays — Signal Dimensions 1946
frame size, specifying 1945
freqaxis 307, 320, 882
freqdivide 107, 125
frequency cuts 77
Frequency Sweep 300, 859
fullpssvec 723
Funnel Cut 83, 123

G

Gain Compression Analysis 97
gear_order 885
gear2 method 671
Guidelines for Setting up Oscillators 694

H

Harmonic Balance 53
harmonic balance
oversample factor 68
Harmonic Balance Method 26
hbac 287
hbac with multiple hb inputs 324
hbnoise 375
hbprecond_solver 323, 398, 885
hbsp 553
highorder 716

I

ic 149, 706
ifft 253
IM1 342
IM2 Distortion Summary 358
IM3 342
indq 1909
inexactNewton 714
input pins, number of in SpectreRF Engine block 1944
Input-referred compression 199
IP2 333
IP3 333
Iteration Limit 102
itres 107, 149, 713

J

Jcc 431
Jee 414, 505
jjitter 393

L

Large signal S Parameters (LSSP) 1695
Large-Signal IP3 272, 799
loadpull 208, 762
lorentzian 399, 524
LSSP amplitude sweep 1704
lteratio 712

M

Math Operations library 1946
MATLAB
design name 1951, 1953
start command 1951
startup directory 1951
MATLAB Support Matrix 1959
maxacfreq 714
Maximum sideband 302, 861
maxiters 708
maxorder 723
maxperiods 148, 715
maxstep 152, 744
Method 711
Model Reuse 1899
Modulated 302
Modulated hbac analysis for oscillator 360
Modulated PAC 862
Modulation Parameters 1771
multiple hbnoise 402, 480
Multiplier 1719
Multi-Sinusoid hb Analysis 276

N

nestlvl 153, 323, 399, 730

© 1994-2020 All Rights Reserved.

netlist
generating for cosimulation 1948
generating for lower-level block in cosimulation 1948
preparing for cosimulation in ADE 1948
preparing for cosimulation without using a GUI 1955
Netlist Driven Flows 38
newlink hbnoise_noise_separation 386
no command value, of Start MATLAB field 1951
noise factor 394
noise figure 394
Noise Parameters 1852, 1883
noise separation 386
noise summary 441
noise temperature 1856
noise type=jitter 408
noisetemp parameter 1856
now command value
in MATLAB cosimulation 1951
of Start MATLAB field 1951
using to test whether ADE starts MATLAB 1951
Nport 1862
Nport Compression 1877

O

One value 1733
oppoint 730
Oscillator Tuning Mode 109, 137, 677, 703
Oscillator Tuning Mode Analysis 234, 831
Oscillator Tuning Mode with Sweeps 238, 835
Oscillator Tuning Range 229
oscsolver 885
Output in Spectre Window 167
output pins, number of in SpectreRF Engine block 1945
outputtype 732
overview, Spectre RF analyses 26

P

PAC 840
PAC analysis 840
PAC with Harmonic Balance PSS 853
PAC with Multiple PSS Inputs 886
PAC with Shooting PSS 855
PAM4 mapping 1810
PAM4 modulation 1810
Parasitic Reduction 30
Period of waveform 1733
periodic AC analysis, *See* PAC analysis
Periodic Analyses 26
periodic steady-state analysis. *See* PSS analysis
periodic transfer function analysis, *See* PXF analysis
pinnode 108, 149
pins, input, in SpectreRF Engine block 1944
pins, output, in SpectreRF Engine block 1945
Plotting Currents 191
Plotting dBm 188
Plotting input impedance magnitude/phase curves 1710
Plotting Kf 574
Plotting loadpull curves 218, 774
Plotting Modulation Sensitivity 829
Plotting Modulation sensitivity 231
Plotting noise figure 576
Plotting noise from sideband -1 461
Plotting Oscillator Output Spectrum 818
Plotting Oscillator Startup Waveform 816
Plotting Output Frequency Vs Tuning Voltage 826
Plotting Output Power Vs Output Frequency 823
Plotting Output Power Vs Tunng Voltage 822
Plotting S11 curve 1706
Plotting Spectral Content 184
Plotting THD 797
Port element 1715
Port number 1719
port parameters 1718
Power Added Efficiency 224
Power gain 789
Power Gain Vs Output Power 795
Power-added efficiency 792
PRBS Model External Triggering 1845
Probe-based Method 109, 140
psaratio 722
PSS 647
PSS analysis 647
Pulse Waveform Examples 1734
Pulse Waveform Parameters 1731

© 1994-2020 All Rights Reserved.

Pulse width 1733
PWL Waveform Parameters 1745
PXF 1117
PXF analysis 1117

Q

Quality of Rational Interpolation 1894
Quasi-Periodic Analyses 28

R

Rapid IP2 305, 333, 865, 894
Rapid IP3 305, 333, 865, 894
Rational Fit Methods 1869
Reactance 1719
readhb 106, 155
readic 150, 707
readns 150, 707
readpss 735
recover 156, 738
related documents 2075
relref 664, 712
resgmrescycle 322, 398, 714, 885
Resistance 1718
restart 163
Rise time 1733
Running the Cosimulation 1957

S

sample time, setting for block used in cosimulation 1945
Sampled 307, 867
Sampled Circuits and Switched-capacitor Filters 881
Sampled hbac analysis for mixer 367
save 153, 323, 398, 730
Save and Recover 105
saveallsidebands 399
saveclock 155, 737
savefile 156, 738
saveperiod 155, 737
savetime 156, 737
Semi--Autonomous 142
Semi-Autonomous 110
Setting Harmonics and Transient Assist Automatically 112
Setting Harmonics Automaticall 66
Setting Harmonics Manually 66
Setup — Matlab/Simulink — Log file 1951
Setup — Matlab/Simulink — Setting 1949
Shooting Method 26
Shooting PSS 675
Simulation — Configuration Parameters 1946
Simulation — Start 1957, 1958
Simulation basics 25
simulation response timeout for cosimulation 1945
Sinusoidal Waveform Parameters 1768
skipcount 731
skipdc 150, 663, 707
skipstart 730
skipstop 731
Small-Signal Parameters 1858
socket mode, for cosimulation 1945
socket port, for cosimulation 1945
solver 884
S-Parameter Equations 1925
S-parameter simulation data, plotting
  equations for S-parameter calculator 1925
Specialized Analyses 298
Specialized Analysis=None 855, 858
Spectre command, for cosimulation 1945
Spectre RF
  analyses, overview 26
SpectreRF Engine
  block, number of input pins in 1944
  block, number of output pins in 1945
  showing port labels for 1945
SpectreRF-workshop 2075
ssbphasenoise 1854
Start MATLAB field
  before Simulation value 1951
  no value 1951
  now value 1951
  values for 1951
steadyratio 713
step 744
stimuli 398
stop time, for cosimulation 1952
strobedelay 731
strobeperiod 731
swapfile 733
Sweeps 124, 191, 780

© 1994-2020 All Rights Reserved.

T

Temperature Effect Parameters 1860
Terminating a port 1716
Three Input Frequencies 81
Three Tone Harmonic Balance 268
tolerance 322, 397, 884
Tones=Frequencies 135
Tones=Names 116, 136
Tools — Analog Environment 1949
Transient Principles 664
Trapezoidal Method 669
Triple Beat 306, 338
tstab 678
tstabmethod 152, 711
tstart 745
Two Input Frequencies 76, 241, 777
typographical conventions 23

U

Up conversion 296, 849
Using Harmonic Port with HB 1719
Using hbac for conversion gain
measurement 311
using in SpectreRF simulation
port parameter types 1718
Using the nport Component 1871

V

View — Simulink library 1946
Viewing total noise 466
Voltage gain 787
Voltage gain compression 202

W

Waveform Entry Method 1747
write 733
writefinal 733
writehb 106, 155
writepss 734

X

xdbmax 163
xdbstart 163
xdbsteps 163
xdbtol 163

Z

Zero value 1733

© 1994-2020 All Rights Reserved.