cādence®

Spectre® Circuit Simulator RF Analysis Library Reference

Product Version 19.1
May 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 © 1979, J. J. Dongarra, J. Du Croz, S. Hammarling, and R. J. Hanson © 1988, J. J. Dongarra, J. Du Croz, I. S. Duff, and S. Hammarling © 1990; University of Tennessee, Knoxville, TN and Oak Ridge National Laboratory, Oak Ridge, TN © 1992-1996; Brian Paul © 1999-2003; M. G. Johnson, Brisbane, Queensland, Australia © 1994; Kenneth S. Kundert and the University of California, 1111 Franklin St., Oakland, CA 94607-5200 © 1985-1988; Hewlett-Packard Company, 3000 Hanover Street, Palo Alto, CA 94304-1185 USA © 1994, Silicon Graphics Computer Systems, Inc., 1140 E. Arques Ave., Sunnyvale, CA 94085 © 1996-1997, Moscow Center for SPARC Technology, Moscow, Russia © 1997; Regents of the University of California, 1111 Franklin St., Oakland, CA 94607-5200 © 1990-1994, Sun Microsystems, Inc., 4150 Network Circle Santa Clara, CA 95054 USA © 1994-2000, Scriptics Corporation, and other parties © 1998-1999; Aladdin Enterprises, 35 Efal St., Kiryat Arye, Petach Tikva, Israel 49511 © 1999 and Jean-loup Gailly and Mark Adler © 1995-2005; RSA Security, Inc., 174 Middlesex Turnpike Bedford, MA 01730 © 2005.

All rights reserved. Associated third party license terms may be found at <install_dir>/doc/OpenSource/*

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 11

Using License Queuing 13

Suspending and Resuming Licenses 13

Related Documents for Spectre 13

Third Party Tools 14

Typographic and Syntax Conventions 15

1

rfLib Library 17

Baseband_components Category 18

BB_driver 20

BB_loss 21

BB_shifter_combiner 22

BB_shifter_splitter 23

BB_xfmr 24

cap_BB 25

dwn_cnvrt 27

HilbertTr_BB 30

ind_BB 31

IQ_demod_BB 33

IQ_mod_BB 36

LNA_BB 39

MPSK 41

PA_BB 44

QPSK 46

Rectangular_16_QAM 49

Rectangular_64_QAM 53

res_BB 57

rfVsourceBB 59

up_cnvrt 61

VGA_BB 64

© 1994-2020 All Rights Reserved.

[Butterworth_filters Category](#) 66
- [BB_butterworth_bp](#) 67
- [BB_butterworth_bs](#) 68
- [BB_butterworth_hp](#) 69
- [BB_butterworth_lp](#) 70
- [butterworth_bp](#) 71
- [butterworth_bs](#) 72
- [butterworth_hp](#) 73
- [butterworth_lp](#) 74
- [Butterworth and Chebyshev Filter Supporting Information](#) 75

[Chebyshev_filters Category](#) 78
- [BB_chebyshev_bp](#) 79
- [BB_chebyshev_bs](#) 80
- [BB_chebyshev_hp](#) 81
- [BB_chebyshev_lp](#) 82
- [chebyshev_bp](#) 83
- [chebyshev_bs](#) 84
- [chebyshev_hp](#) 85
- [chebyshev_lp](#) 86

[Measurements Category](#) 87
- [CDMA_reverse_xmit](#) 88
- [comms_instr](#) 92
- [eye_diagram_generator](#) 95
- [gmsk](#) 98
- [GSM_xmtr](#) 99
- [instr_term](#) 102
- [offset_comms_instr](#) 103
- [pi_over4_dqpsk](#) 104
- [polar_rect](#) 108
- [rect_polar](#) 109

[Passband_components Category](#) 110
- [IQ_demodulator](#) 111
- [IQ_modulator](#) 113
- [LNA_PB](#) 116
- [MIXER_PB](#) 118
- [PA_PB](#) 120

© 1994-2020 All Rights Reserved.

shifter_combiner 121
shifter_splitter 122
RF_components Category 123
ACPR_source 124
balun 125
balun_com 127
gfsk 128
lna 131
mixer 135
ofdm 139
osc 141
pa 144
quadrature 147
RFVsource 149
shifter 151
Triplexer 153
Ten_plexer 155
Testbenches Category 159
AM_PM_test_ckt 160
ava_pwr_gain 165
BB_ind_cap_test 169
demod_ip3 175
dwn_cnvt_test 183
mixer_ip3 188
mod_1dbcp 195
mod_demod_test 198
noise_figure 199
one_db_cp 201
PB_BB_filter_comparison 203
PB_ind_cap_test 204
quad_and_phase_error_demo 205
shifter_combiner_test 207
shifter_splitter_test 214
up_cnvt_test 218
view_switching 220
WCDMA_components Category 221

© 1994-2020 All Rights Reserved.

wcdma_dl_com_chanl . . . 222
wcdma_ocns . . . 223
wcdma_power_adjust . . . 224
wcdma_qpsk . . . 225
wcdma_sch_multiplexer . . . 227
wcdma_scrambling . . . 228
wcdma_scr_generator . . . 229
wcdma_spreading . . . 230
Wireless Components Category . . . 231
LTE . . . 232
NR . . . 236
ZIGBEE . . . 239
smartmeter . . . 242
DMR . . . 246
WLAN11A . . . 249
WLAN11P . . . 252
WLAN11N . . . 256
WLAN11AC . . . 261
WLAN11AD . . . 266
WLAN11AF . . . 271
WLAN11AH . . . 276
WLAN11AX . . . 280
Bluetooth . . . 284
WPROBE . . . 290
Modifying the BB Signal Generators Using Modelwriter . . . 294

2
rfTlineLib Library . . . 299
Introduction to RFTLINELIB . . . 300
RFTLINELIB Benefits . . . 301
RFTLINELIB Cells Vs. analogLib Mtline . . . 301
Installation . . . 302
Location . . . 302
Library Structure . . . 303
Quick-Start Guide . . . 304

© 1994-2020 All Rights Reserved.

Using RFTLINELIB 310
- The Stackup 310
- Dielectric Models 310
- Adding Dielectric Material Models 316
- Conductors 316
- Adding Conductor Materials 318
- Configuring Layers 318
- Entering Custom Layer Stacks 319
- Inserting Layers 319
- Bindkeys 320
- Units 320
- Adding Transmission Lines 321
- Working With Transmission Line Discontinuities 322

RFTLINELIB Cells 322
- SYMBOL: MLIN 322
- SYMBOL: SLIN 324
- SYMBOL: MCLIN 325
- SYMBOL: MACLIN 326
- SYMBOL: SCLIN 328
- SYMBOL: NCLIN 329
- SYMBOL: STACKUP 333
- SYMBOL: MTEE 334
- SYMBOL: STEE 341
- SYMBOL: MSTEP 348
- SYMBOL: SSTEP 349
- SYMBOL: MBEND2 350
- SYMBOL: SBEND2 352
- SYMBOL: MBEND 354
- SYMBOL: SBEND 356
- SYMBOL: MCORN 358
- SYMBOL: MCROS 359
- SYMBOL: SCROS 363
- SYMBOL: MCURVE 366
- SYMBOL: SCURVE 368
- SYMBOL: MLOCEFF 370
- SYMBOL: SLOCEFF 371

© 1994-2020 All Rights Reserved.

SYMBOL: CLIN 372
SYMBOL: TLINP 375
Obsolete Components 379
Microstrip Components 380
microbend2 381
microbend90 383
microopenend 385
microstep 387
microstrip 389
Stripline Components 391
stripbend90 392
stripline 394

3
Introduction to the PLL library 397
Models in the PLL library 398
Introduction to the PLL Library Documentation 399
Phase-Domain Model of a Simple PLL 399
Example 1: Dynamic Test for Capture Range and Lock Range 404
Example 2: Loop Gain Measurement 405
Example 3: PM Input 423
Modeling a PFD-Based PLL 425
VCO 426
Charge Pump 427
Loop Filter 427
State-Space Averaged PFD (Phase-Domain Phase-Frequency Detector Model) . . 428
Lock Indicator 431
Example 5: Comparison With a Voltage-Domain Model 433
How the PFD Model Works 435
How the PDF/CP Pump Works 435
References 440

© 1994-2020 All Rights Reserved.

Preface

Spectre® circuit simulator or Spectre® Accelerated Parallel Simulator (Spectre APS) base products with the Spectre(R) 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(R) RF analysis option 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 2-1 on page 11 and Table 2-2 on page 12 show the capabilities offered with the base products Spectre circuit simulator or Spectre APS and the Spectre(R) RF analysis option.

Table 2-1 Spectre® Simulator with Spectre(R) 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)
Envelope Analysis (AM, PM, FM, Autonomous)
Rapid Distortion Analyses - Perturbation-based IP2 and IP3
Co-simulation with Simulink® from The MathWorks
MMSIM Toolbox for MATLAB® from The MathWorks

Table 2-2 Spectre® Accelerated Parallel Simulator with Spectre(R) 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)
Envelope Analysis (AM, PM, FM, Autonomous)
Rapid Distortion Analyses - Perturbation-based IP2 and IP3

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

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

© 1994-2020 All Rights Reserved.

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*.

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 SpectreRF and related products.

© 1994-2020 All Rights Reserved.

- 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 Reference 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 Virtuoso 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].

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.

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

© 1994-2020 All Rights Reserved.

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.
`. . .`	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.

© 1994-2020
All Rights Reserved.

1

rfLib Library

The elements contained in the RF library, `rfLib`, are organized into the following categories:

Categories	Description of Category and Link
`Everything`	Lists all elements in `rfLib`.
`Baseband_components`	"Baseband_components Category" on page 18
`Butterworth_filters`	"Butterworth_filters Category" on page 66
`Chebyshev_filters`	"Chebyshev_filters Category" on page 78
`Measurements`	"Measurements Category" on page 87
`Passband_components`	"Passband_components Category" on page 110.
`RF_components`	"RF_components Category" on page 123
`Testbenches`	"Testbenches Category" on page 159
`WCDMA_components`	"WCDMA_components Category" on page 221
`Wireless`	"Wireless Components Category" on page 231

The `rfLib` elements support the design of both RF circuits and RF systems.

This chapter also describes how to use the Modelwriter to modify the baseband signal generators. For information, see "Modifying the BB Signal Generators Using Modelwriter" on page 294.

© 1994-2020 All Rights Reserved.

Baseband_components Category

The `Baseband_components` category contains the top-down baseband models of common architectural function blocks. The default view of these models is the baseband view (called `veriloga`) but most models in this category also have a differential passband view (called `veriloga_PB`). The `BB_loss` and `VGA_BB` models are exceptions because they are meant only for baseband analysis and have no passband view.

The `Baseband_components` models provide a fast method of mapping RF system specifications into detailed RF designs. The baseband models facilitate fast evaluation of candidate RF architectures specified with DSP metrics.

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 modeling approach taken in top-down design is to simulate only those effects that drive the decisions at hand.

Baseband modeling does not replace passband modeling because some effects missed by equivalent baseband models can affect high level decisions. However, the application of baseband models early followed by passband models later minimizes the number of slow simulations needed at low levels of design abstraction. Baseband models help you to quickly weed out designs that would surely fail tests simulated with passband models.

The cells in the `Baseband_components` category are:

- BB_driver
- BB_loss
- BB_shifter_combiner
- BB_shifter_splitter
- BB_xfmr
- cap_BB
- dwn_cnvrt
- HilbertTr_BB
- ind_BB
- IQ_demod_BB
- IQ_mod_BB

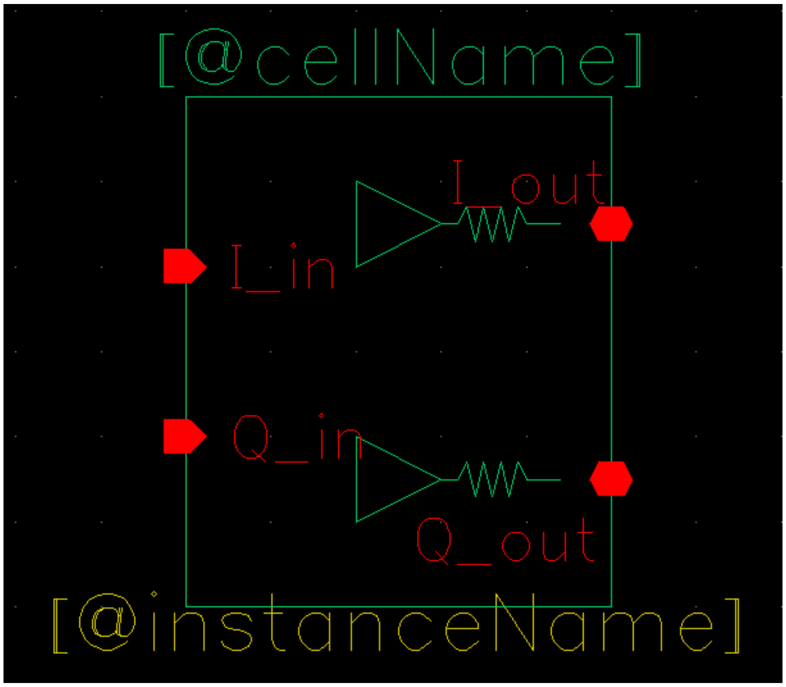
© 1994-2020 All Rights Reserved.

- LNA_BB
- PA_BB
- res_BB
- rfVsourceBB
- up_cnvrt
- VGA_BB

© 1994-2020
All Rights Reserved.

BB_driver

(Baseband Driver)



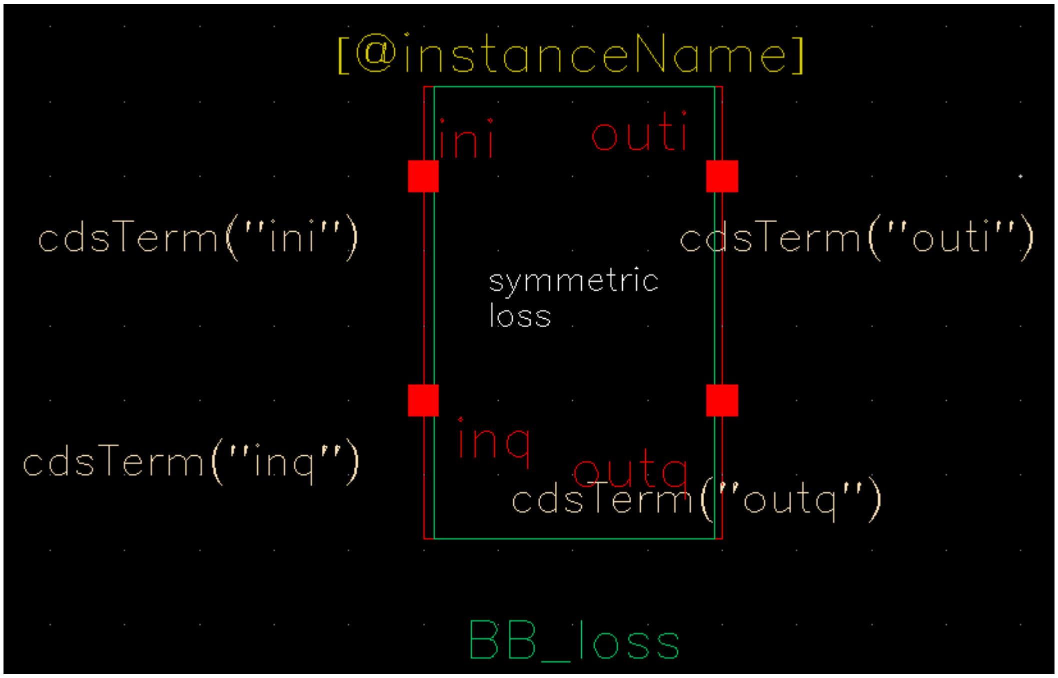
The BB_driver element senses a baseband voltage signal and amplifies it.

The parameters are:

`rout`	Output impedance.
`power_level`	

© 1994-2020 All Rights Reserved.

BB_loss



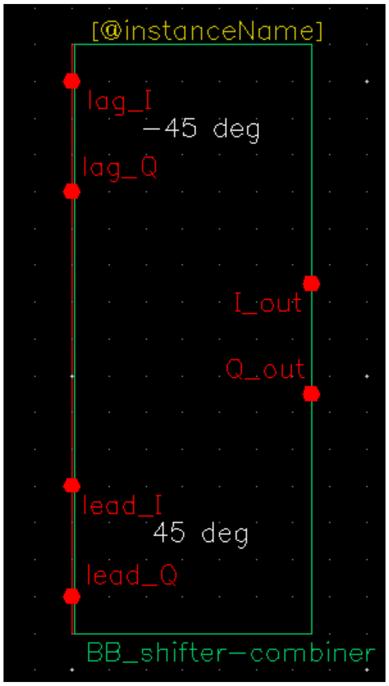
The `BB_loss` element is designed to be used with error vector magnitude (EVM) calculations. EVM is defined in terms of an ideal receiver or transmitter. If you want to remove a filter's response from the ideal receiver model while leaving only the passband attenuation, replace the filter with a `BB_loss` element and give it the same insertion loss as the filter. There is no passband view or counterpart for this model.

The parameters are:

`loss`	Filter insertion loss [dB].
`r2`	Reference impedance at port 2 [ohm].

© 1994-2020 All Rights Reserved.

BB_shifter_combiner

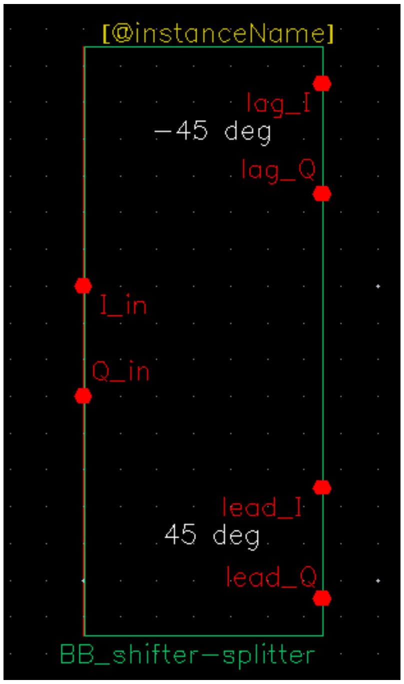


The parameters are:

`fcr`	Carrier frequency.
`freq`	Frequency at which the magnitudes are balanced.
`gain`	Linear scale factor that multiplies the input voltage.
`r`	Impedance of the internal resistor.
`rin`	Input terminal impedances.
`rout`	Output impedance.

© 1994-2020 All Rights Reserved.

BB_shifter_splitter



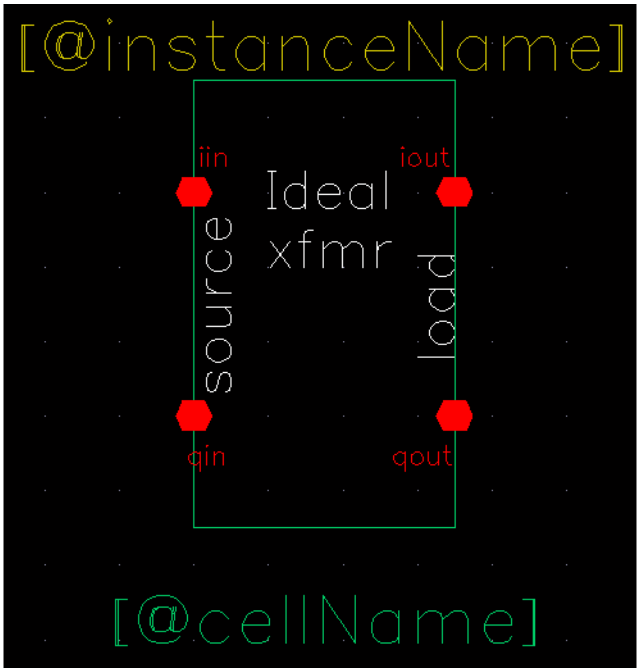
The parameters are:

`fcr`	Carrier frequency.
`freq`	Frequency at which the magnitudes are balanced.
`gain`	Linear scale factor that multiplies the input voltage.
`r`	Impedance of the internal resistor.
`rin`	Input terminal impedances.
`rout`	Output impedance.

© 1994-2020 All Rights Reserved.

BB_xfmr

(Ideal Transformer)



The purpose of the ideal transformer is to help designers transform between different resistances.

The parameters are:

`rsource`	`Rs`, source resistance
`rload`	`RI`, load resistance

- Inputs: i and q input voltages, i and q output currents.
- Outputs: i and q output voltages, i and q input currents, defined as:

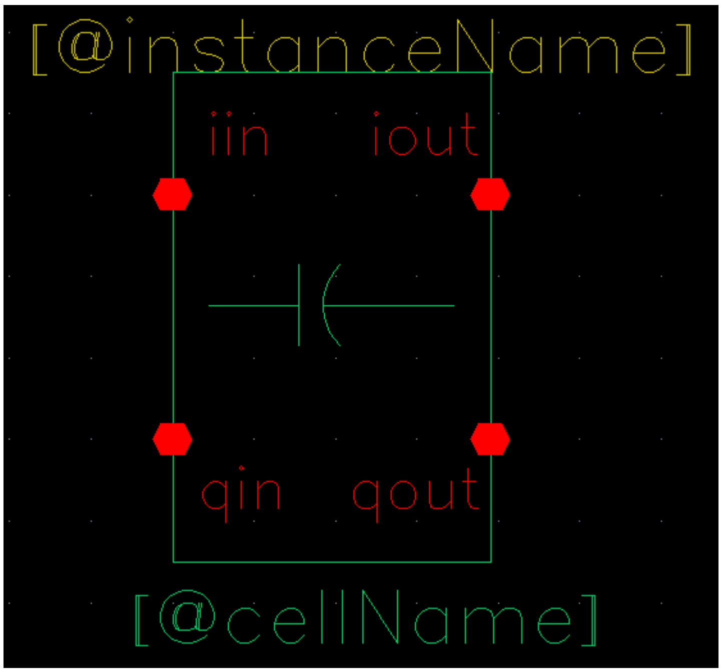
$$\begin{bmatrix} i_{iin} \\ i_{qin} \end{bmatrix} = \sqrt{\frac{Rl}{Rs}} \begin{bmatrix} i_{iout} \\ i_{qout} \end{bmatrix}$$

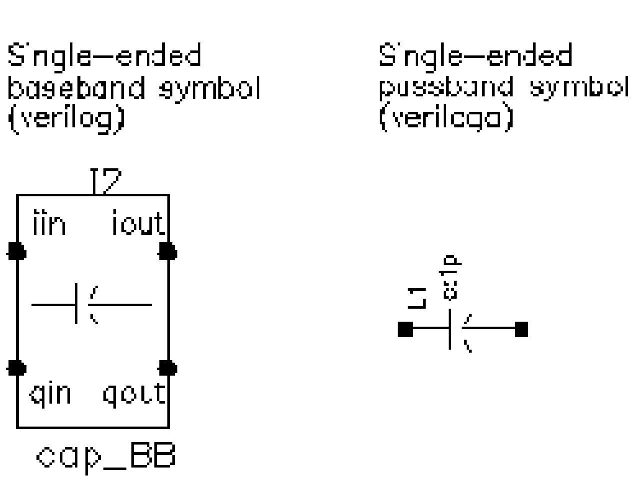
$$\begin{bmatrix} v_{iout} \\ v_{qout} \end{bmatrix} = \sqrt{\frac{Rl}{Rs}} \begin{bmatrix} v_{iin} \\ v_{qin} \end{bmatrix}$$

© 1994-2020 All Rights Reserved.

cap_BB

Figure 1-1 Circuit

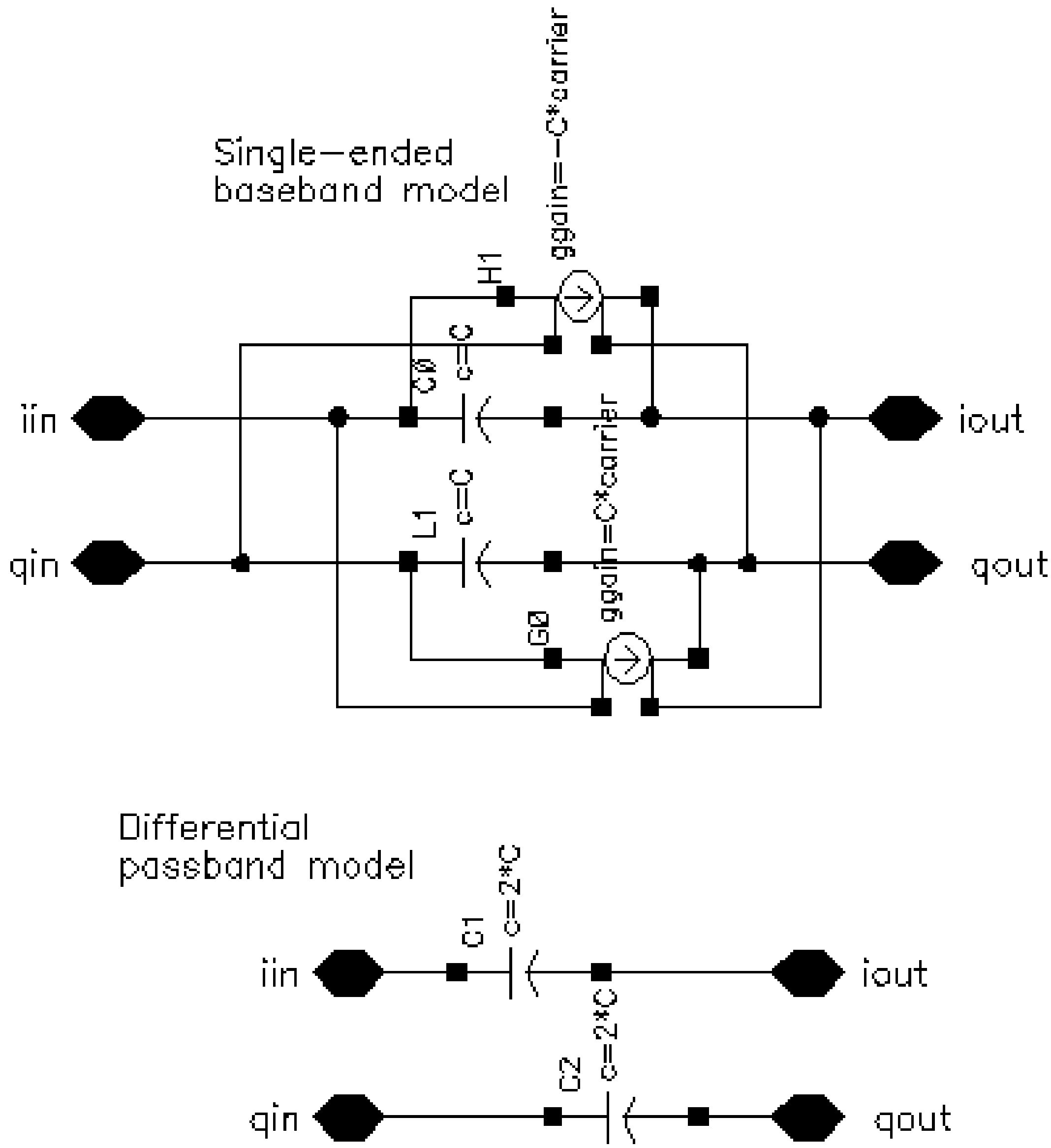




The capacitor is the mathematical dual of the inductor. Figure 1-2 shows the baseband and differential passband capacitor models.

© 1994-2020
All Rights Reserved.

Figure 1-2 Capacitor Model



The parameters are:

`cap`

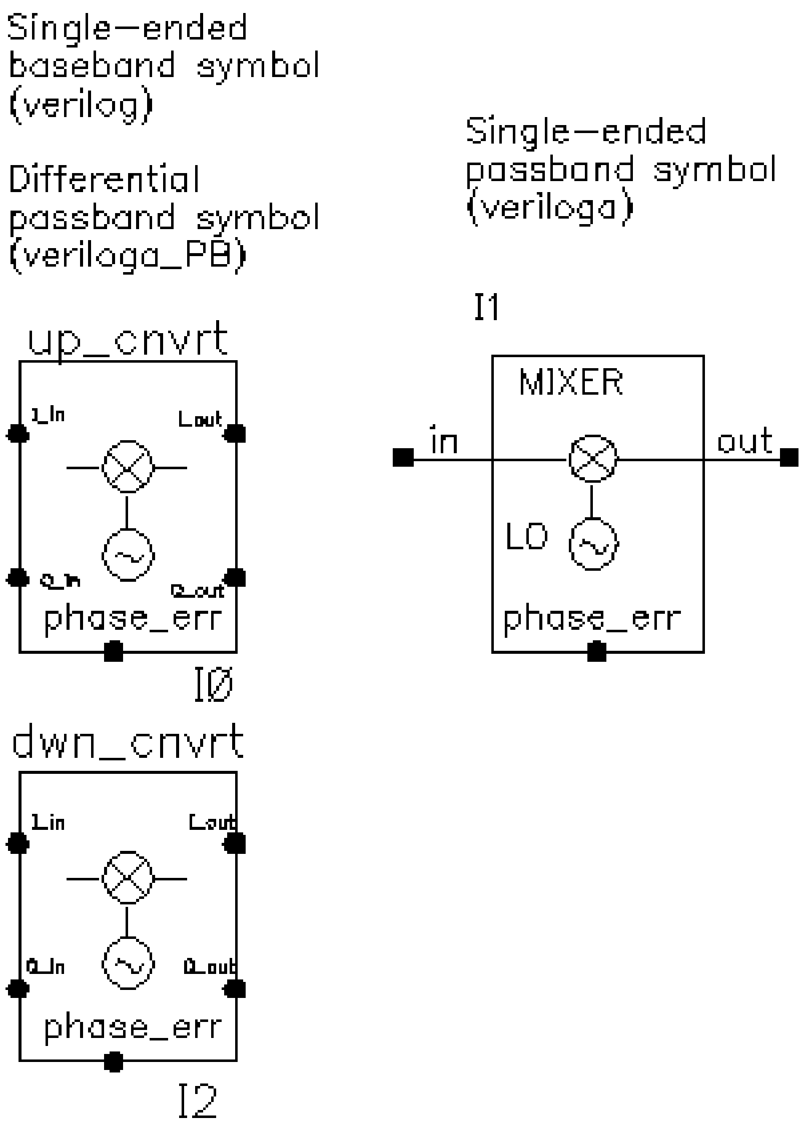
`carrier`

© 1994-2020 All Rights Reserved.

dwn_cnvrt

`(baseband = dwn_cnvrt)`

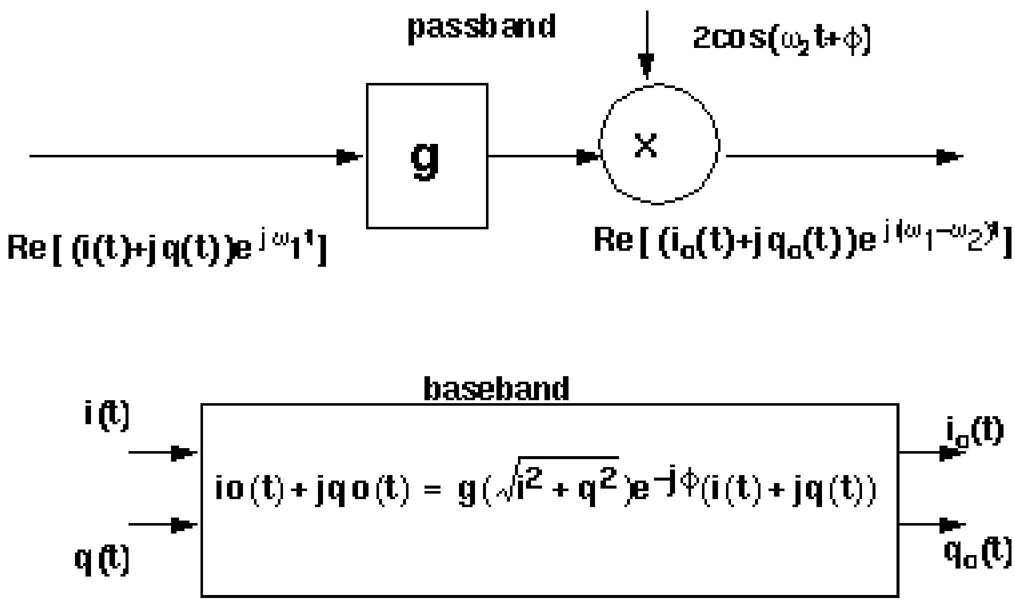
Figure 1-3 Baseband and Passband Mixer Models



`dwn_cnvrt` model is a baseband equivalent model of a mixer used to convert from RF to IF. There are some minor differences in the baseband models that depend on whether conversion is up or down. Figure 1-4 shows what the model does.

© 1994-2020 All Rights Reserved.

Figure 1-4 Calculations for dwn_cnvrt Mixer



The parameters are:

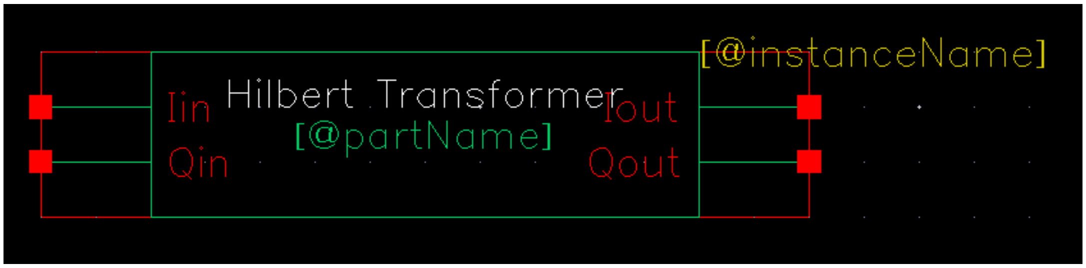
`cmp`	Input power pt for phase point [dBm].
`cw`	{1,0,-1} for {cw, none, ccw}. Defines the direction of the phase shift. 1 for clockwise, 0 for no phase shift, -1 for counter clockwise.
`flo`	Local oscillator frequency.
`frf`	RF frequency.
`gain`	Available power gain [dB].
`IP3`	Input referenced IP3 [dbm].
`nf`	Noise figure.
`pscp`	|radians|@cmp. Defines the absolute value of the output phase shift at the 1dB compression point for power amplifiers. This is the phase shift at an arbitrary output power level for some models.
`psinf`	|radians|@ big input. Defines the absolute value of the output phase shift as input power goes to infinity (if it could go to infinity).
`rin`	Input resistance.
`rout`	Output resistance.

© 1994-2020
All Rights Reserved.

`shp`	AM/PM sharpness. Determines how fast the phase shift occurs with increasing input power. A larger number delays the shift but makes the shift rise faster as a function of input signal level.

© 1994-2020 All Rights Reserved.

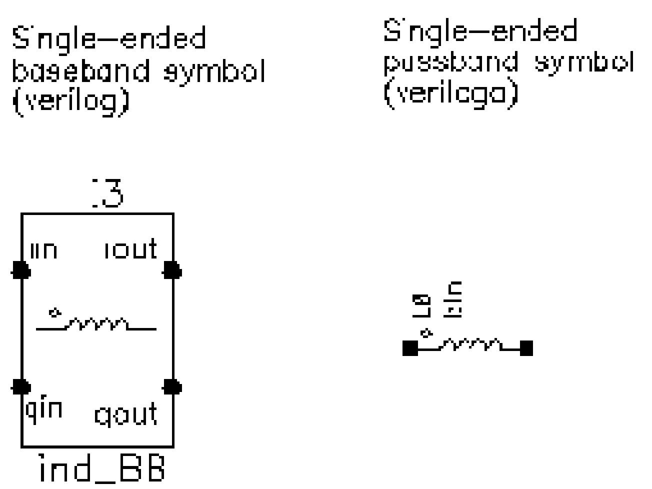
HilbertTr_BB



© 1994-2020
All Rights Reserved.

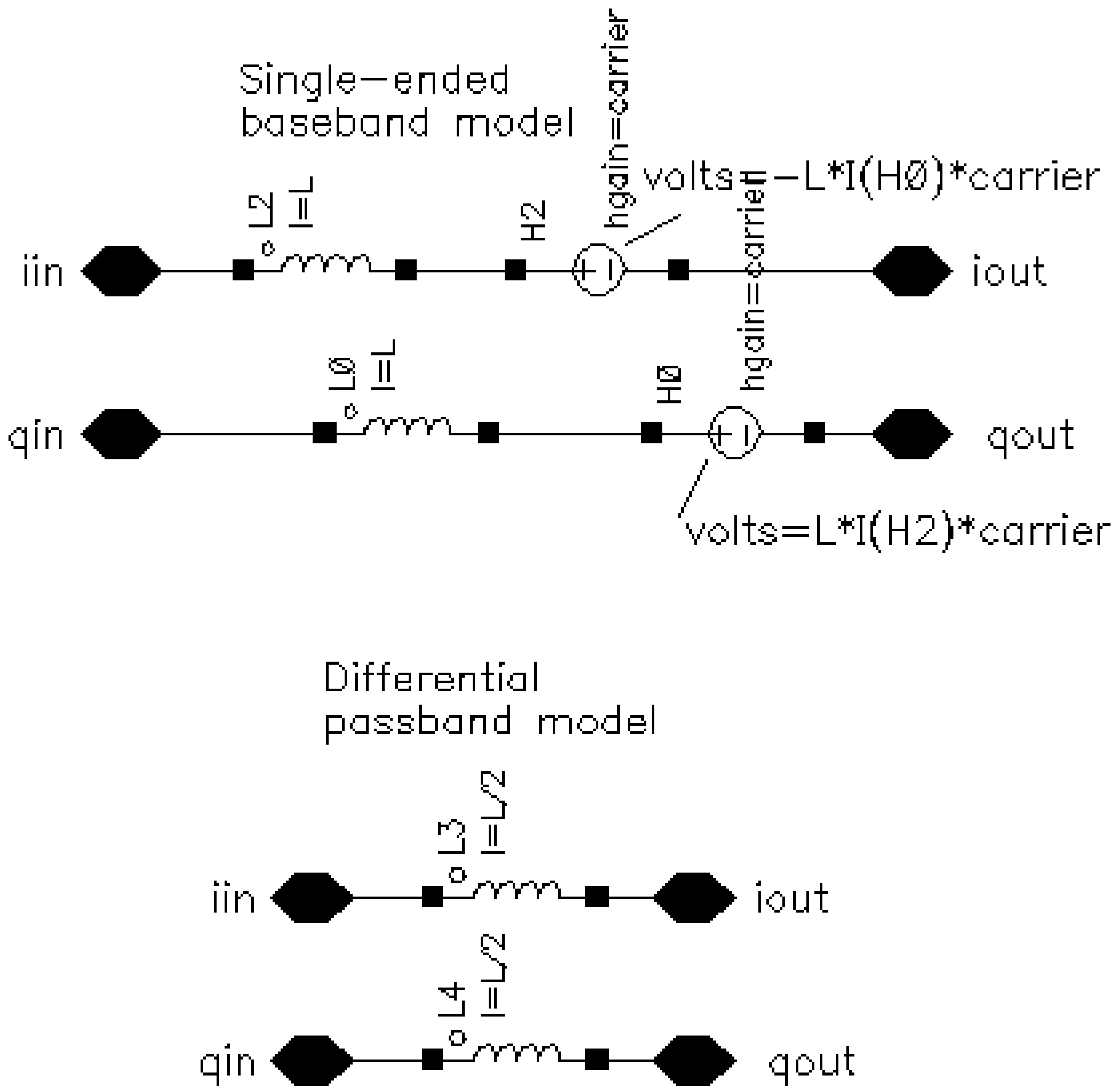
ind_BB

Figure 1-5 Circuit



The baseband inductor model requires one additional parameter besides the inductance, the carrier frequency. Figure 1-6 shows equivalent schematics of the baseband and differential passband inductor models. The inductor models are noiseless.

Figure 1-6 Inductor Model



The parameters are:

`carrier`

`ind`

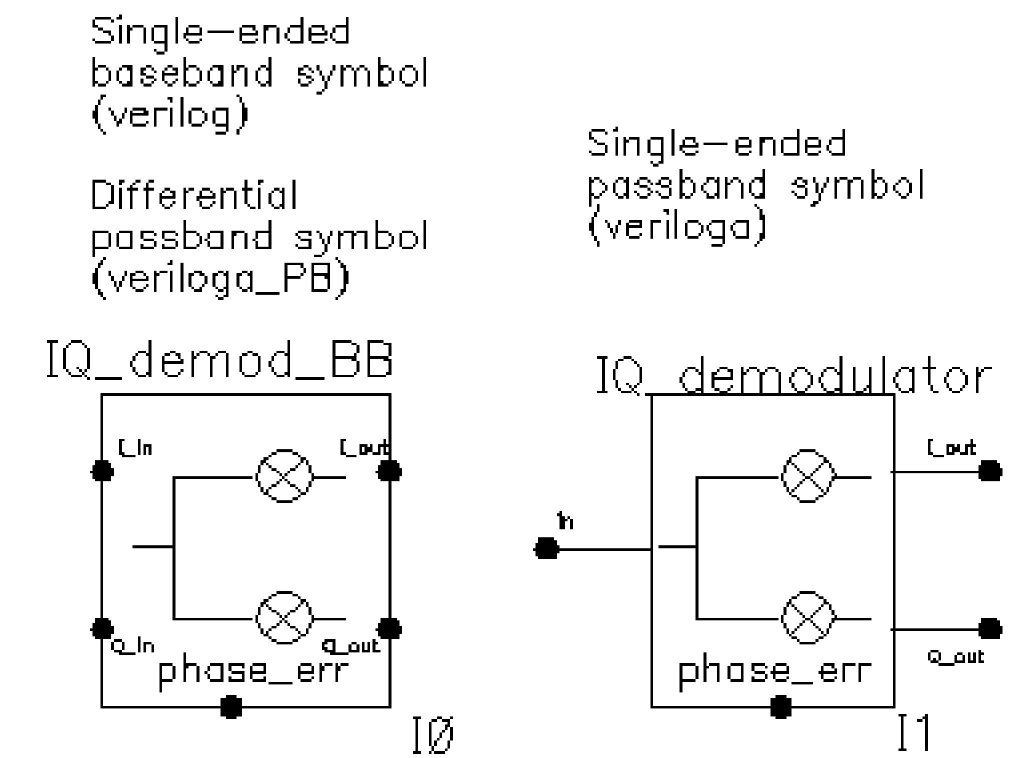
© 1994-2020 All Rights Reserved.

IQ_demod_BB

(IQ Demodulator)

`(baseband = IQ_demod_BB)`

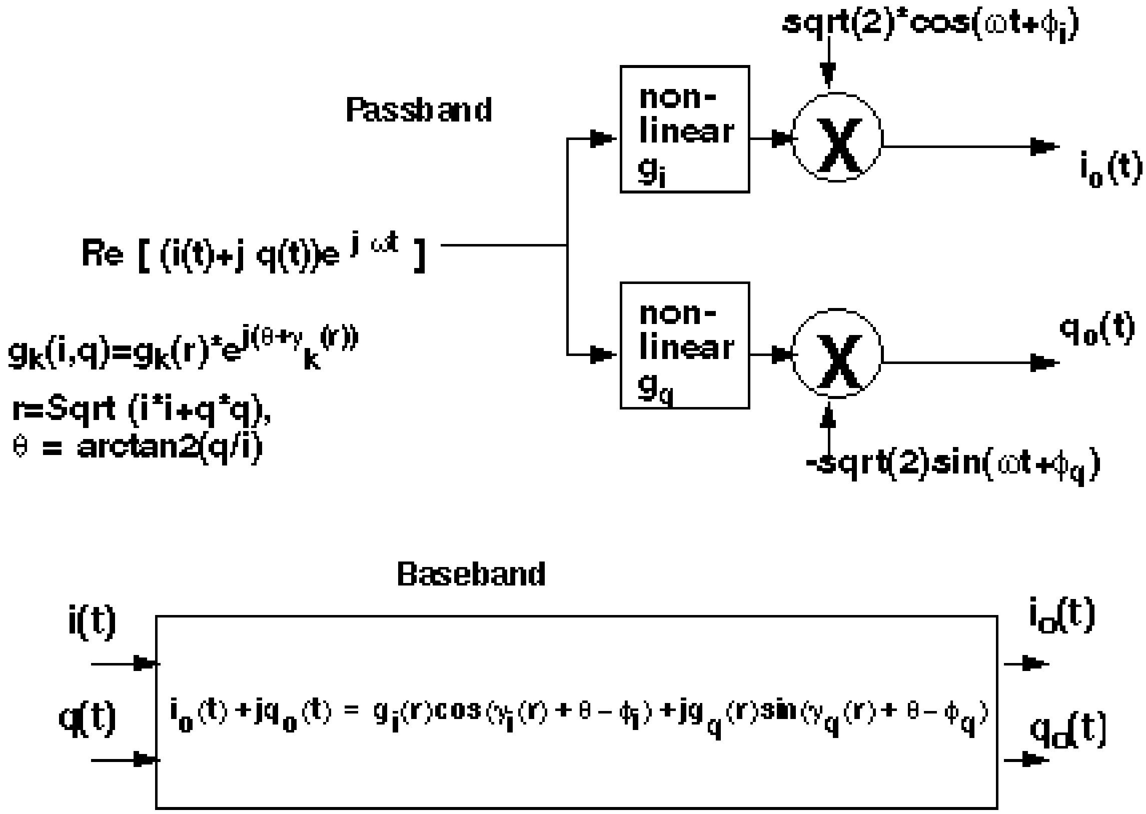
Figure 1-7 Baseband and Passband IQ Demodulator Models



The `IQ_demod_BB` converts RF (or IF) to baseband. Figure 1-8 shows exactly what the passband demodulator model does. The parameters are like those in the modulator blocks except saturation is specified by input referred IP3 instead of by 1 dB compression point. IP3 was chosen over the 1 dB compression point for specifying saturation because the demodulator usually lies in the receive path and receiver blocks are usually specified with IP3.

© 1994-2020 All Rights Reserved.

Figure 1-8 IQ Demodulator Calculations



The parameters are:

`I_cmp`	Input power point for phase point [dBm].
`I_cw`	Determines the direction of the phase shift. The phase shift is only in one direction. +1 means counter-clockwise, -1 means clockwise, and 0 means no phase shift (no am/pm conversion). {1,0,-1} for {cw, none, ccw}.
`I_gain`	Available power gain [dB].
`I_IP3`	Input referenced IP3 [dbm].
`I_pscp`	Output phase shift at cmp [radians]. I-radians@I_cmp.
`I_psinf`	Output phase shift as the input power goes to infinity. I-radians@big I-input.

© 1994-2020 All Rights Reserved.

`I_shp`	Determines how fast the phase shift occurs with increasing input power. A larger number delays the shift but makes the shift rise faster as a function of input signal level. I-sharpness factor.
`nf`	Noise figure [dB].
`Q_cmp`	Input power point for phase point [dBm].
`Q_cw`	Determines the direction of the phase shift. The phase shift is only in one direction. +1 means counter-clockwise, -1 means clockwise, and 0 means no phase shift (no am/pm conversion). {1,0,-1} for {cw, none, ccw}.
`Q_gain`	Voltage gain [dB].
`Q_IP3`	Input referenced IP3 [dBm].
`Q_pscp`	Output phase shift at cmp [radians]. Q-radians@Q_cmp.
`Q_psinf`	Output phase shift as the input power goes to infinity.
`Q_shp`	Determines how fast the phase shift occurs with increasing input power. A larger number delays the shift but makes the shift rise faster as a function of input signal level. Q-sharpness factor.
`rin`	Input resistance.
`rout`	Output resistance.

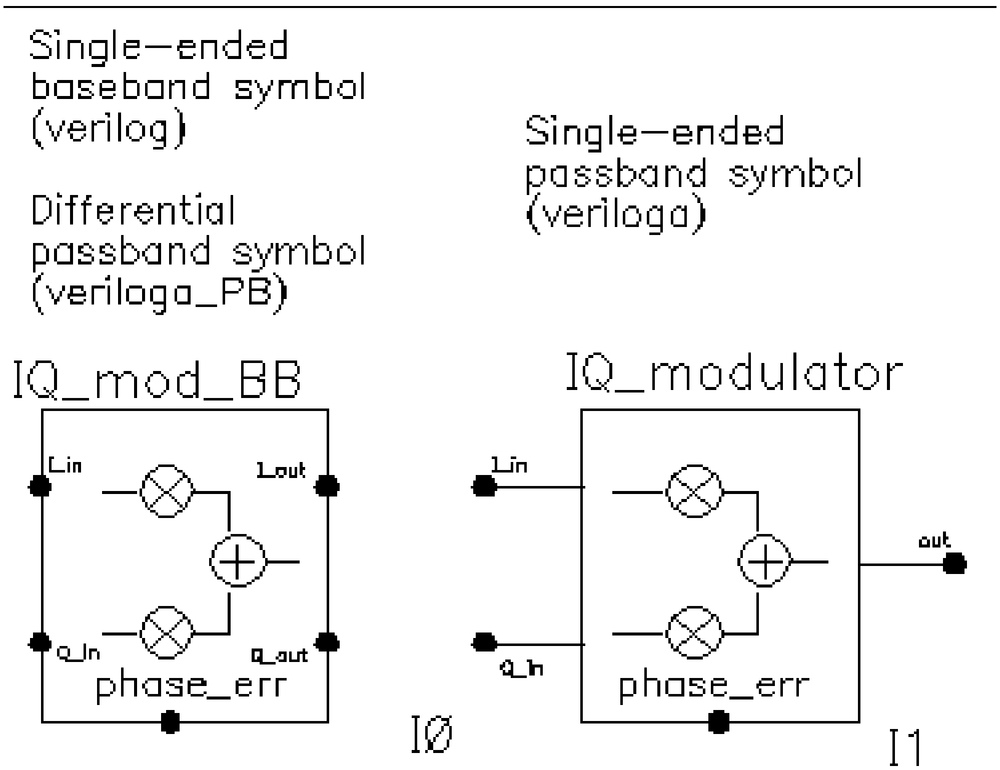
© 1994-2020 All Rights Reserved.

IQ_mod_BB

(IQ Modulator Models)

`(baseband = IQ_mod_BB)`

Figure 1-9 Baseband and Passband IQ Modulator Models



The `IQ_mod_BB` converts baseband signals to RF or IF. Figure 1-10 summarizes exactly what the passband IQ modulator model does. The only difference between the baseband and passband models is carrier suppression. The non-linear functions, g_i and g_q, are specified by their available power gain and 1dB compression points just as in the power amplifier. The functions γ_i and γ_q characterize AM/PM effects in each mixer and are specified by the same parameters that specify power amplifier AM/PM conversion. Because noise is always added at the input, and the input is at baseband in this case, the noise sources are not doubled as they are in the power amplifier or LNA models. Noise figure is defined with reference to one input. Noise is injected at both inputs but the noise injected at just one input alone produces the specified noise figure. Thus, the noise figure parameter should be interpreted as noise figure per input. This model also includes a parameter called `quadrature error` which specifies how far away the two local oscillators signals are from being exactly in quadrature.

© 1994-2020
All Rights Reserved.

`Phase error` is the voltage on the phase error pin. The phase error pin has a fixed noiseless resistive input impedance of 50 ohms. The phase error pin can be used to introduce a dynamic phase error or phase noise. Phase noise can be fed into the phase error pin from a phase-domain PLL model or from a Port. Noise in Port models can be specified either by the internal resistance or by a data file that tabulates a power spectral density. The phase error pin can also be driven by a ramp or circular integrator output to model a frequency error between the incoming carrier and local oscillator.

The following parameters specify the IQ modulator. The available power gain and one dB compression point are explained first. The effects of the phase_error pin and the quadrature error parameter are discussed at the end of this section.

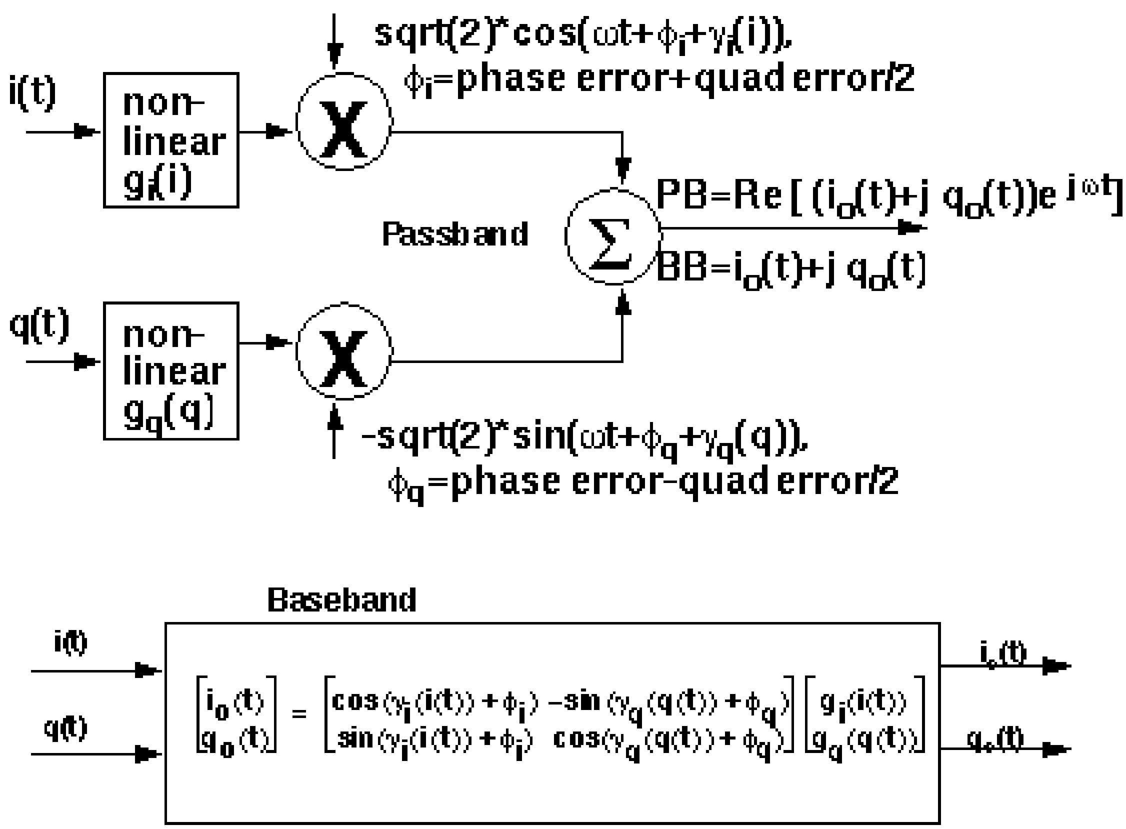
The parameters are:

`I_cmp`	Input power point for phase point [dBm].
`I_cpdb`	I db compression point, referred to the output. I-output 1dB CP [dBm].
`I_cw`	Determines the direction of the phase shift. The phase shift is only in one direction. +1 means counter-clockwise, -1 means clockwise, and 0 means no phase shift (no am/pm conversion). {1,0,-1} for {cw, none, ccw}.
`I_gain`	Available power gain [dB].
`I_pscp`	Output phase shift at cmp [radians]. I-radians@I_cmp.
`I_psinf`	Output phase shift as the input power goes to infinity. I-radians@big I-input.
`I_shp`	Determines how fast the phase shift occurs with increasing input power. A larger number delays the shift but makes the shift rise faster as a function of input signal level. I-sharpness factor.
`nf`	Noise figure [dB].
`Q_cmp`	Input power point for phase point [dBm].
`Q_cw`	Determines the direction of the phase shift. The phase shift is only in one direction. +1 means counter-clockwise, -1 means clockwise, and 0 means no phase shift (no am/pm conversion). {1,0,-1} for {cw, none, ccw.
`Q_gain`	Voltage gain [dB].
`Q_IP3`	Input referenced IP3 [dBm].
`Q_pscp`	Output phase shift at cmp [radians]. Q-radians@Q_cmp.

© 1994-2020 All Rights Reserved.

`Q_psinf`	Output phase shift as the input power goes to infinity.
`Q_shp`	Determines how fast the phase shift occurs with increasing input power. A larger number delays the shift but makes the shift rise faster as a function of input signal level. Q-sharpness factor.
`rin`	Input resistance.
`rout`	Output resistance.

Figure 1-10 IQ Modulator Calculations



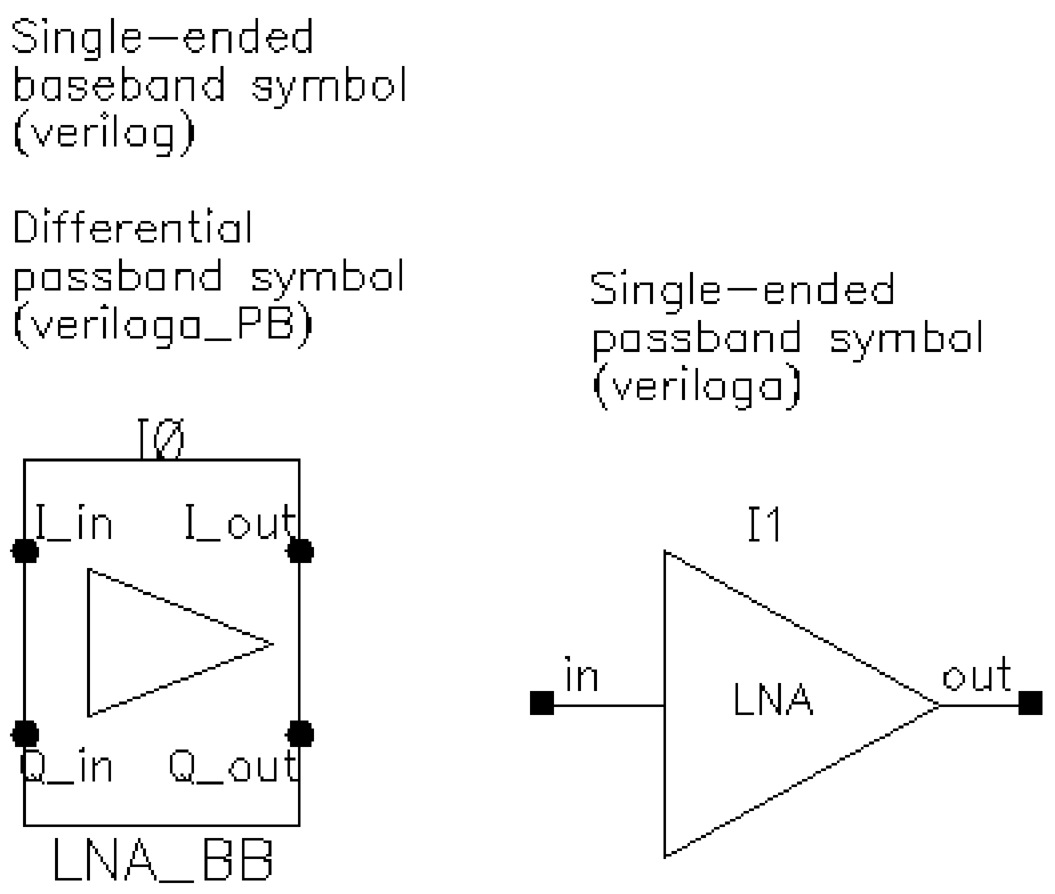
© 1994-2020
All Rights Reserved.

LNA_BB

(Low Noise Amplifier Models)

`(baseband = LNA_BB)`

Figure 1-11 Baseband and Passband Power Amplifier Models



The parameters are:

`cmp`	cmp [dBm]. Output power level where the next parameter is defined.
`cw`	{1,0,-1} for {cw, none, ccw}. Defines the direction of the phase shift. 1 for clockwise, 0 for no phase shift, -1 for counter clockwise.
`gain`	Available power gain [dB].
`IP3`	Input referred IP3 [dBm].

© 1994-2020
All Rights Reserved.

`nf`	Noise figure [dB].
`pscp`	|radians|@cmp. Defines the absolute value of the output phase shift at the 1dB compression point for power amplifiers. This is the phase shift at an arbitrary output power level for some models.
`psinf`	|radians|@ big input. Defines the absolute value of the output phase shift as input power goes to infinity (if it could go to infinity).
`rin`	Input resistance.
`rout`	Output resistance.
`shp`	AM/PM sharpness. Defines how steep the output phase shift changes are with respect to input power.

© 1994-2020 All Rights Reserved.

MPSK

MPSK takes in a bit signal or generates a pseudo-random bit stream internally and then generates an unfiltered I and Q modulating signal at the output. The number of PSK states is eight, and the output is generated as points on a unit circle.

The properties are:

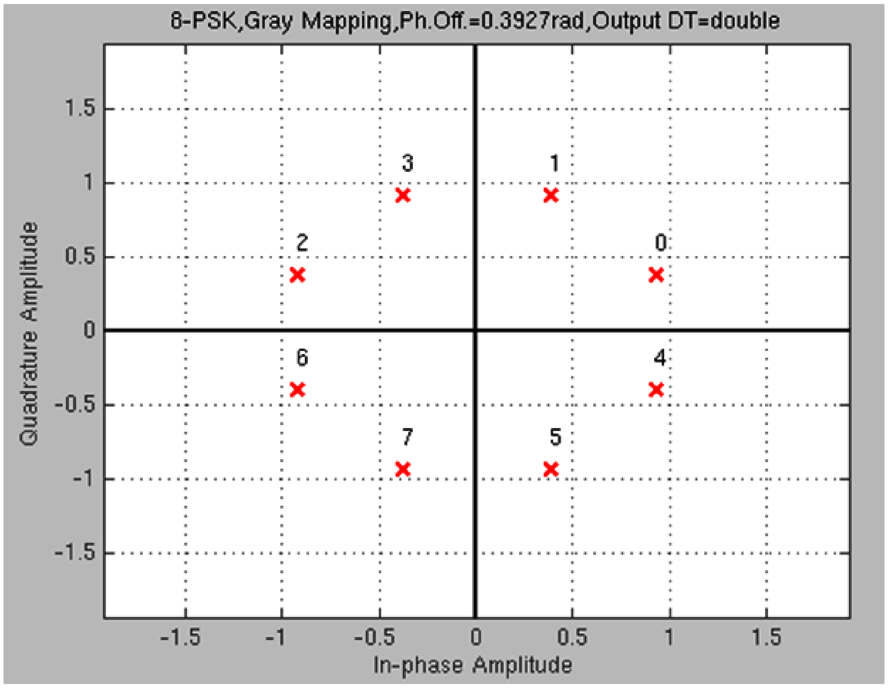
© 1994-2020
All Rights Reserved.

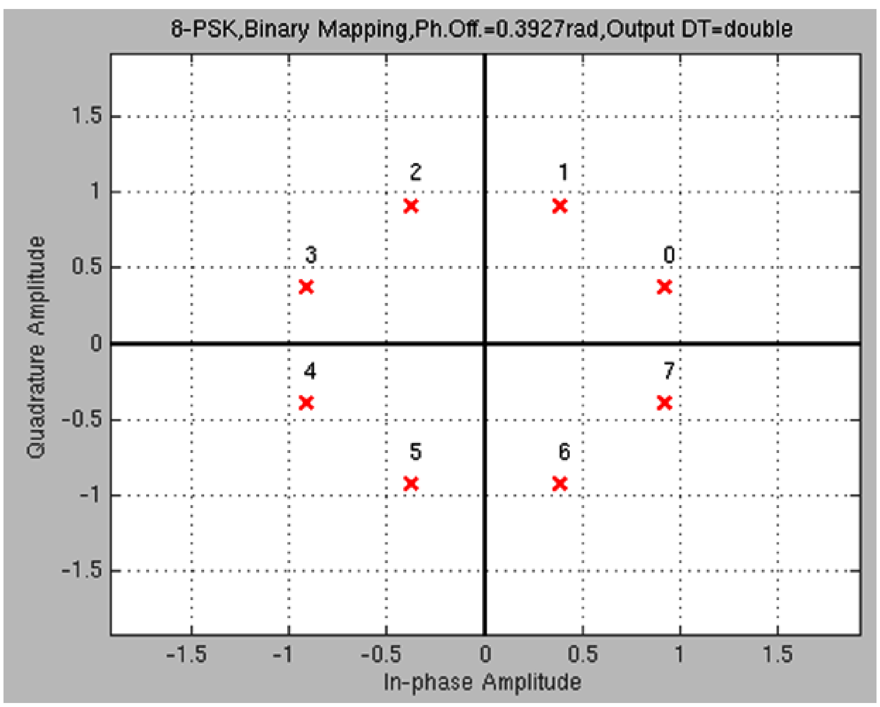
Note: If you are using the MPSK block in the schematic capture tool and you use the properties form, select `veriloga` as the CDF Parameter of view.

`Symbol_Period`	Defines the period of the transitions of the I and Q signal at the output
`Constellation_Ordering`	Legal values are `Gray`, `Binary`, and `user-defined`. ■ `Gray` generates phase states separated by a one bit change from constellation point to point. ■ `Binary` generates 000 at the first point, 001 at the second point, and so on. ■ `user_defined` generates the list provided in the user_mapping_vec at each point. If the user_mapping_vec is 0,7,1,6,2,5,3,4, then the first point is 000, the second point is 111, the third point is 001, and so on.
`Usr_mapping_vec`	Defines the binary digit for each succeeding constellation point when `Constellation_Ordering` is set to `user_defined`. The list must be enclosed in square brackets. The list must include zero. To generate the 8 phase states, the numbers 0 through 7 need to be entered in the list. If the user_mapping_vec is 0,7,1,6,2,5,3,4, then the first point is 000, the second point is 111, the third point is 001, and so on.
`phase_offset`	Defines the angle of the first point in the constellation. The number entered is multiplied by pi internally.
`Disable_input`	0 causes the input to be read and the output I and Q generated based on the input vector at the start of the symbol period. 1 causes an internal pseudo-random number to be generated at the beginning of each symbol period.
`Seed`	This is the seed for the internal pseudo-random bit generator.

© 1994-2020 All Rights Reserved.

Figure 1-12 Constellation Map For MPSK





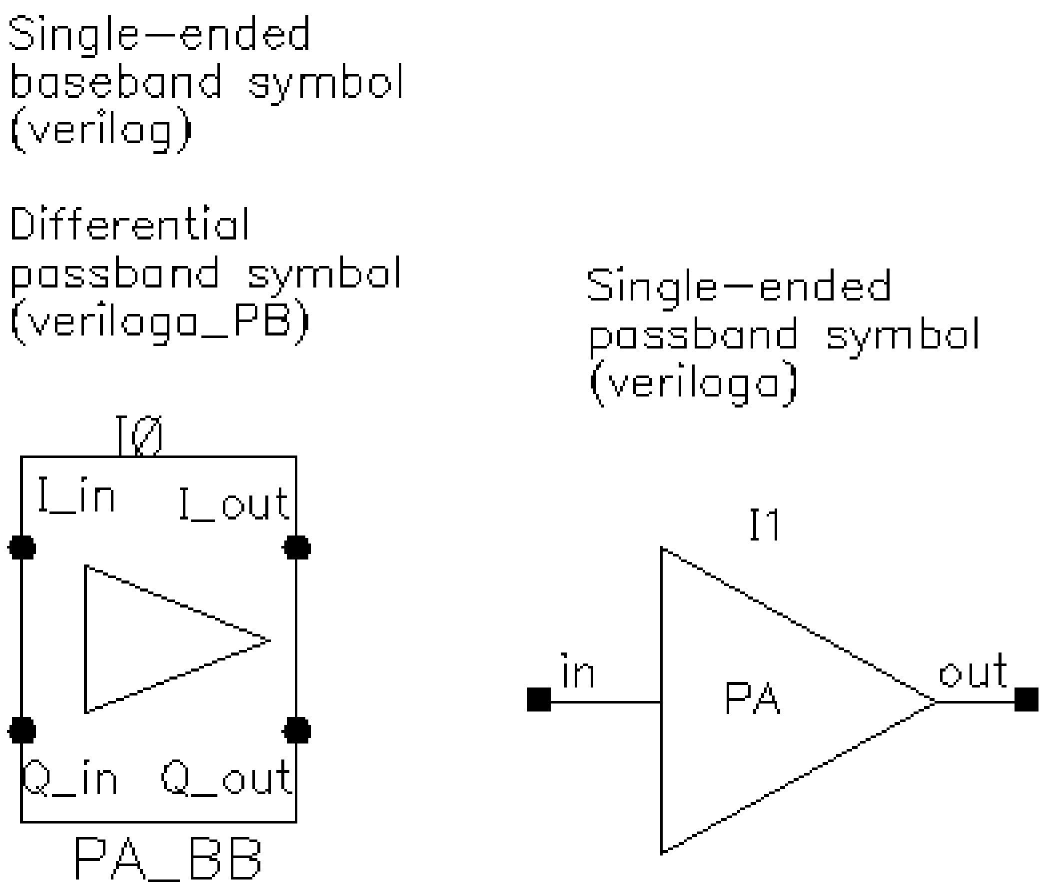
© 1994-2020 All Rights Reserved.

PA_BB

(Power Amplifier Model)

`(baseband = PA_BB)`

Figure 1-13 Baseband and Passband Power Amplifier Models



The following parameters specify the power amplifier model.

The parameters are:

`cpdb`	Output 1db cp [dBm].
`cw`	{1,0,-1} for {cw, none, ccw}. Defines the direction of the phase shift. 1 for clockwise, 0 for no phase shift, -1 for counter clockwise.
`gain`	Available power gain.

© 1994-2020
All Rights Reserved.

`nf`	Noise figure.
`pscp`	|radians|@cmp. Defines the absolute value of the output phase shift at the 1dB compression point for power amplifiers. This is the phase shift at an arbitrary output power level for some models.
`psinf`	|radians|@ big input. Defines the absolute value of the output phase shift as input power goes to infinity (if it could go to infinity).
`rin`	Input resistance.
`rout`	Output resistance.
`shp`	AM/PM sharpness. Defines how steep the output phase shift changes are with respect to input power.

© 1994-2020
All Rights Reserved.

QPSK

QPSK takes in a bit signal or generates a pseudo-random bit stream internally and then generates an unfiltered I and Q modulating signal at the output. The number of PSK states is four and the output is generated as points on a unit circle.

The properties are:

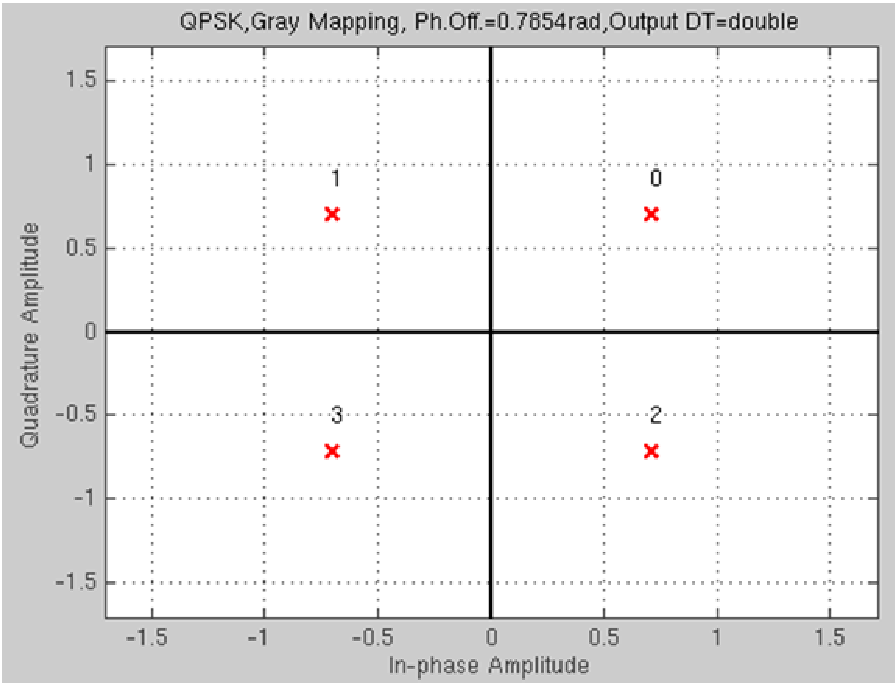
© 1994-2020
All Rights Reserved.

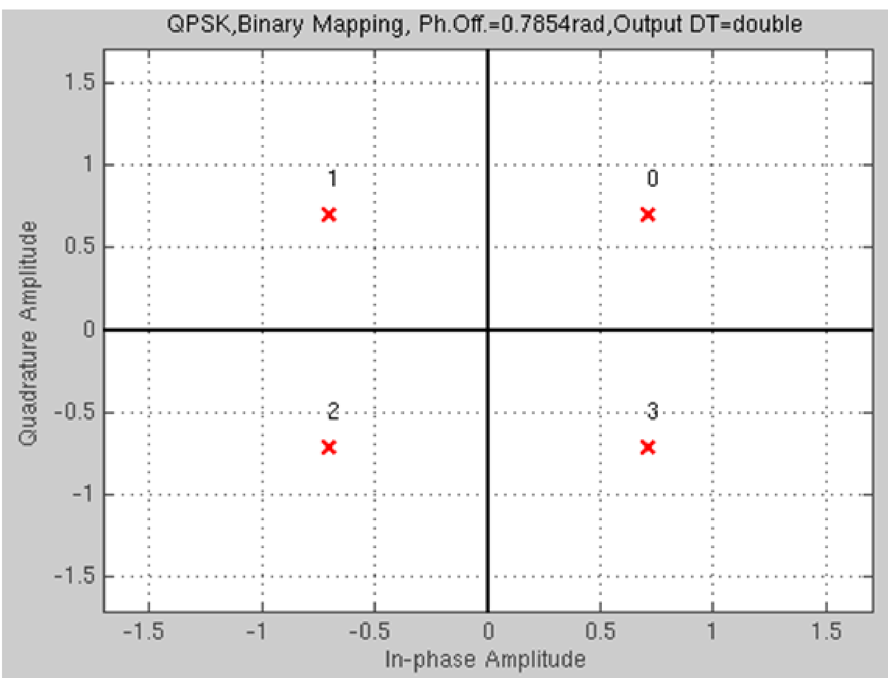
Note: If you are using the QPSK block in the schematic capture tool and you use the properties form, select `veriloga` as the CDF Parameter of view.

`Symbol_Period`	Defines the period of the transitions of the I and Q signal at the output
`Constellation_Ordering`	Legal values are Gray, Binary, and user-defined. ■ `Gray` generates phase states separated by a one bit change from constellation point to point. ■ `Binary` generates 00 at the first point, 01 at the second point, and so on. ■ `user_defined` generates the list provided in the `user_mapping_vec` at each point. If the `user_mapping_vec` is 0,3,1, 2, then the first point is 00, the second point is 11, the third point is 01, and so on.
`Usr_mapping_vec`	Defines the binary digit for each succeeding constellation point. The list must be enclosed in square brackets. The list must include zero. To generate the 4 phase states, the numbers 0 through 3 need to be entered in the list. The order controls the bit representation of each phase state when `Constellation_Ordering` is set to `user_defined`. If the `user_mapping_vec` is 0,3,1,2, then the first point is 00, the second point is 11, the third point is 01, and so on.
`phase_offset`	Defines the angle of the first point in the constellation. The number entered is multiplied by `pi` internally.
`Disable_input`	0 causes the input to be read and the output I and Q generated based on the input vector at the start of the symbol period. 1 causes an internal pseudo-random number to be generated at the beginning of each symbol period.
`Seed`	This is the seed for the internal pseudo-random bit generator.

© 1994-2020
All Rights Reserved.

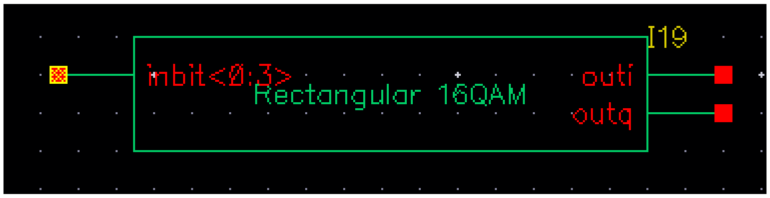
Figure 1-14 Constellation Map For QPSK





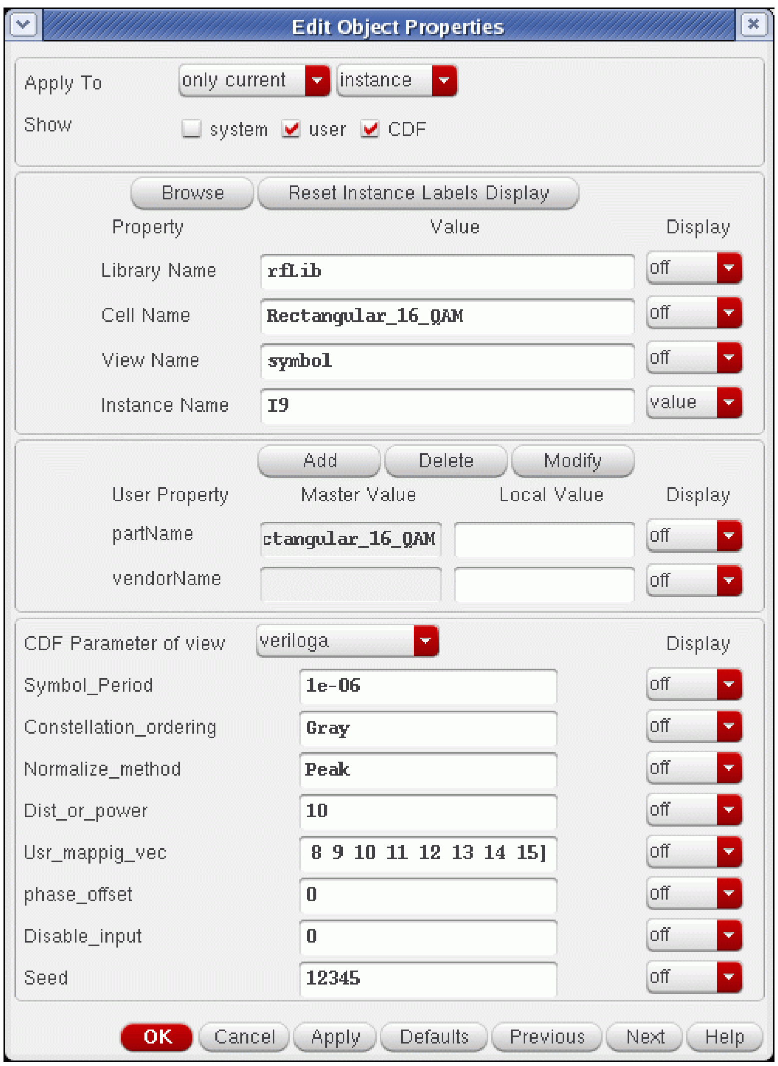
© 1994-2020 All Rights Reserved.

Rectangular_16_QAM



Rectangular_16_QAM takes in a bit signal or generates a pseudo-random bit stream internally and then generates an unfiltered I and Q modulating signal at the output. The number of QAM states is 16 and the output is generated as 16-QAM points separated by the voltages specified by the Dist_or_power property by default.

The properties are:



© 1994-2020 All Rights Reserved.

Note: If you are using the Rectangular_16_QAM block in the schematic capture tool and you use the properties form, select veriloga as the CDF Parameter of view.

Symbol_Period	Defines the period of the transitions of the I and Q signal at the output
Constellation_Ordering	Legal values are Gray, Binary, and user-defined. ■ Gray generates phase states separated by a one bit change from constellation point to point. ■ Binary generates 0000 at the first point, 0001 at the second point, and so on. ■ user_defined generates the list provided in the user_mapping_vec at each point. If the user_mapping_vec is 0,15, 1, 14, 2, 13, 3, 12, 4, 11, 5, 10, 6, 9, 7, 8, then the first point is 0000, the second point is 1111, the third point is 0001, and so on.
Normalize_method	The default is min distance. This is the minimum distance in volts andbetween points in the constellation diagram. With the default of 2, the Dist_or_powerpoints is generated with 2 volt distances between points. For example, a point at 1,1 is generated, and 1,3, 1,5, and so on, are also generated. When Normalize_method is set to peak, the power in watts for a 1 ohm load at the largest amplitude point of the constellation (the corners) is entered in the Dist_or_power property When the Normalize_method is set to Average, the average power in watts for a 1 ohm load is specified in the Dist_or_power property.

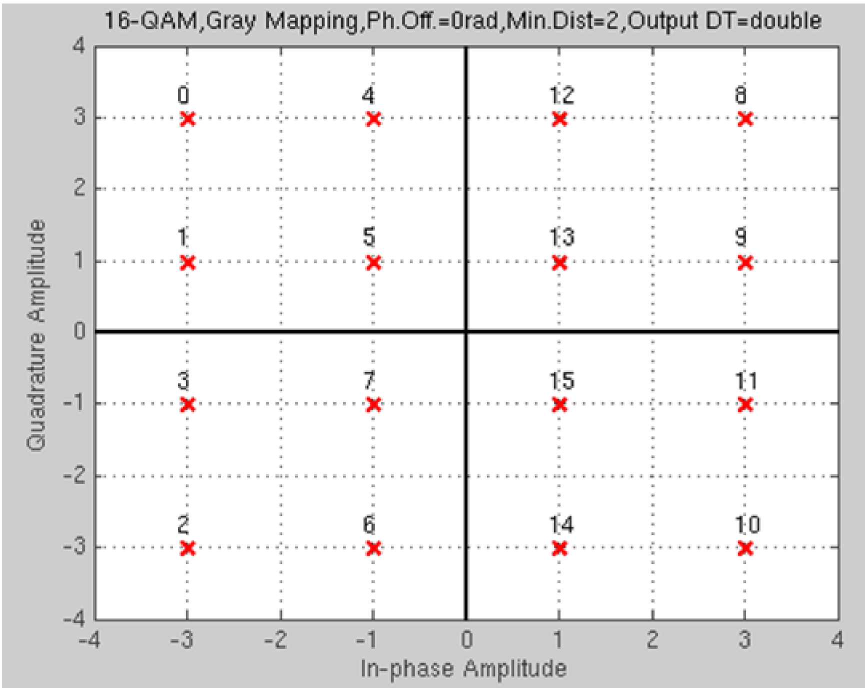
© 1994-2020

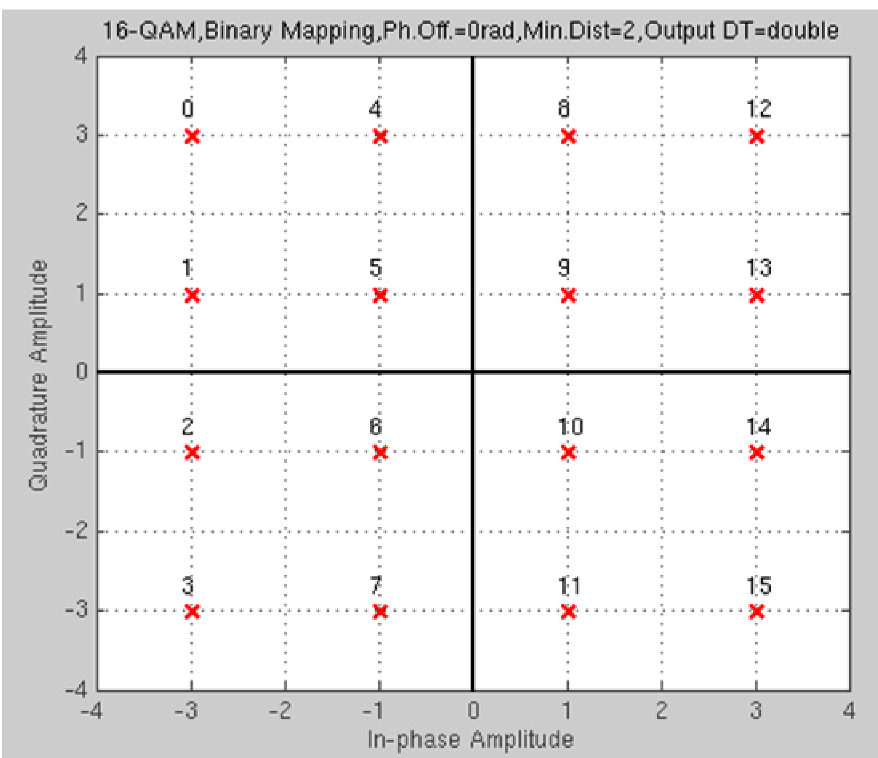
All Rights Reserved.

`Usr_mapping_vec`	Defines the binary digit for each succeeding constellation point. The list must be enclosed in square brackets and must include zero. To generate the 16 phase states, the numbers 0 through 15 need to be entered in the list. The order controls the bit representation of each phase state when `Constellation_Ordering` is set to `user_defined`. If the `user_mapping_vec` is 0, 15, 1, 14, 2, 13, 3, 12, 4, 11, 5, 10, 6, 9, 7, 8, then the first point is 0000, the second point is 1111, the third point is 0001, and so on.
`phase_offset`	Defines the angle of the first point in the constellation. The number entered is multiplied by pi internally.
`Disable_input`	0 causes the input to be read and the output I and Q generated based on the input vector at the start of the symbol period. 1 causes an internal pseudo-random number to be generated at the beginning of each symbol period.
`Seed`	This is the seed for the internal pseudo-random bit generator.

© 1994-2020
All Rights Reserved.

Figure 1-15 Constellation Map For Rectangular_16_QAM



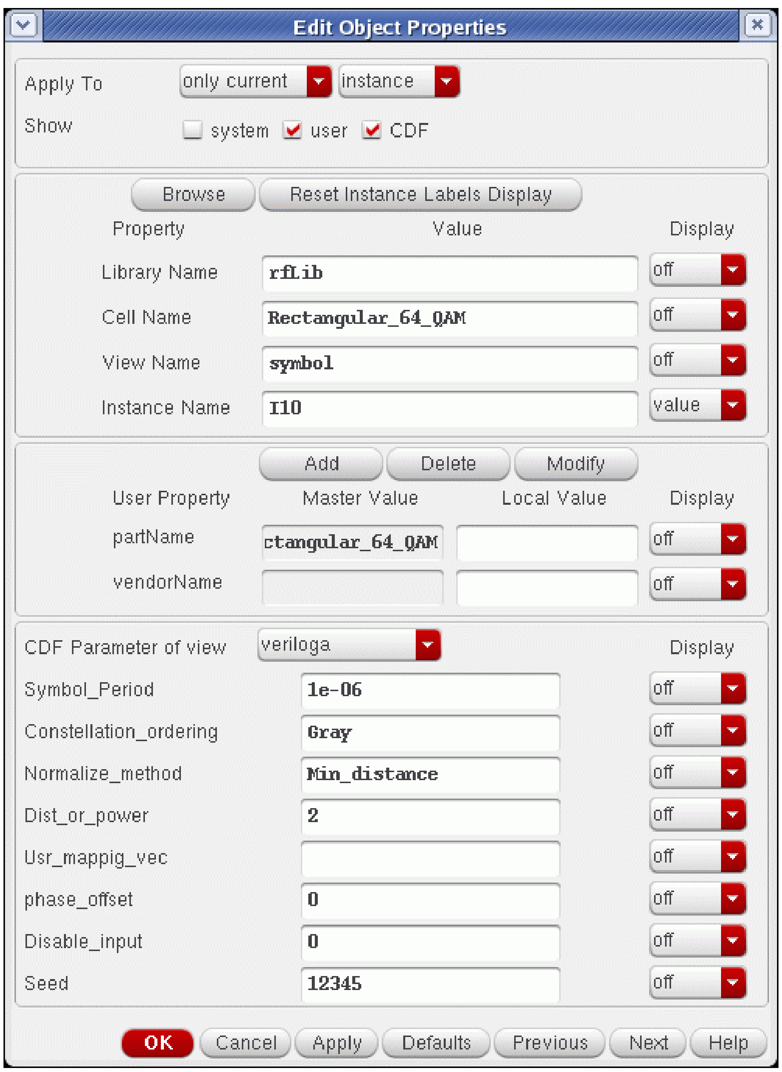


© 1994-2020
All Rights Reserved.

Rectangular_64_QAM

`Rectangular_64_QAM` takes in a bit signal or generates a pseudo-random bit stream internally and then generates an unfiltered I and Q modulating signal at the output. The number of QAM states is 64 and the output is generated as 64-QAM points separated by the voltages specified by the `Dist_or_power` property by default.

The properties are:



© 1994-2020
All Rights Reserved.

Note: If you are using the `Rectangular_64_QAM` block in the schematic capture tool and you use the properties form, select `veriloga` as the CDF Parameter of view.

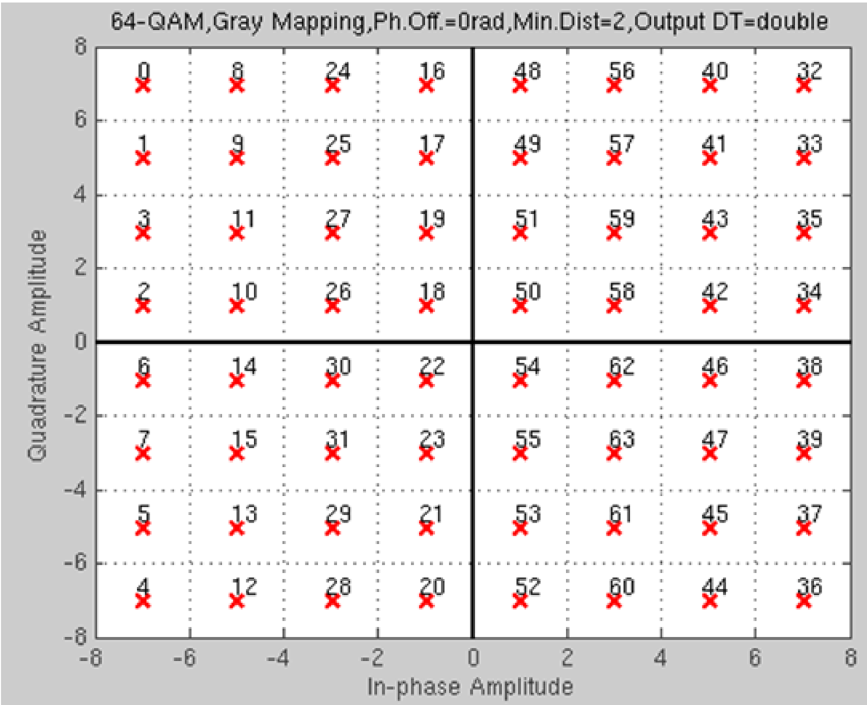
`Symbol_Period`	Defines the period of the transitions of the I and Q signal at the output
`Constellation_Ordering`	Legal values are `Gray`, `Binary`, and `user-defined`. ■ `Gray` generates phase states separated by a one bit change from constellation point to point. ■ `Binary` generates 000000 at the first point, 000001 at the second point, and so on. ■ `user_defined` generates the list provided in the user_mapping_vec at each point. If the `user_mapping_vec` is 0,63, 1, 62, … , 31, 32, then the first point is 000000, the second point is 111111, the third point is 000001, and so on.
`Normalize_method`	The default is `min` distance. This is the minimum distance in volts andbetween points in the constellation diagram. With the default of 2, the `Dist_or_powerpoints` is generated with 2 volt distances between points. For example, a point at 1,1 is generated, and 1,3, 1,5, and so on are also generated. When `Normalize_method` is set to peak, the power in watts for a 1 ohm load at the largest amplitude point of the constellation (the corners) is entered in the `Dist_or_power` property When the `Normalize_method` is set to `Average`, the average power in watts for a 1 ohm load is specified in the `Dist_or_power` property.
`Usr_mapping_vec`	Defines the binary digit for each succeeding constellation point when `Constellation_Ordering` is set to `user_defined`. The list must be enclosed in square brackets and must include zero. To generate the 64 phase states, the numbers 0 through 63 need to be entered in the list. If the `user_mapping_vec` is 0, 63, 1, 62, … , 31, 32, then the first point is 000000, the second point is 111111, the third point is 000001, and so on.
`phase_offset`	Defines the angle of the first point in the constellation. The number entered is multiplied by pi internally.

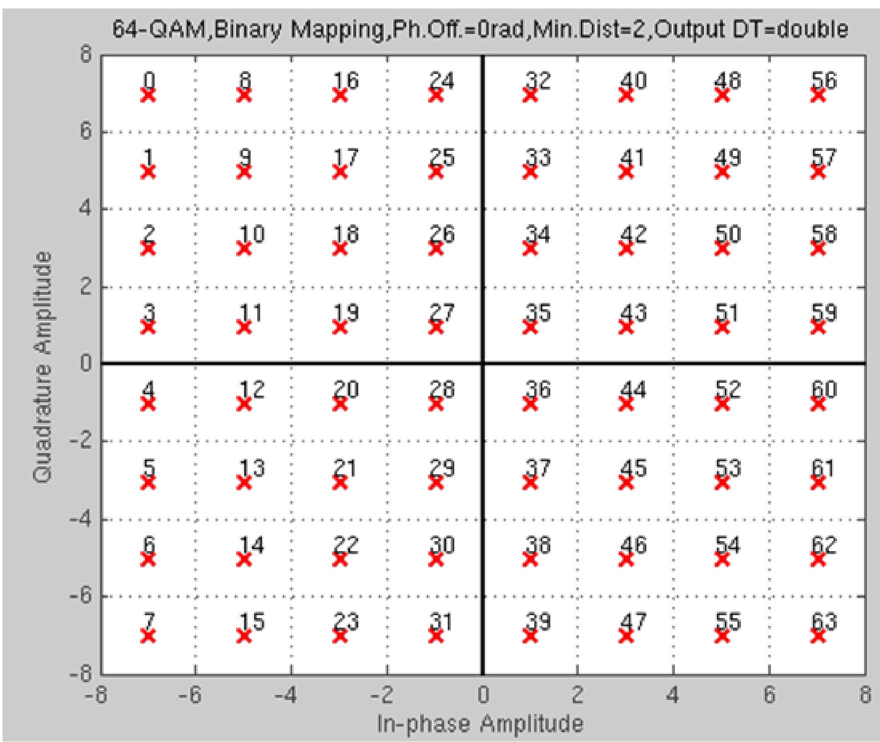
© 1994-2020 All Rights Reserved.

`Disable_input`	0 causes the input to be read and the output I and Q generated based on the input vector at the start of the symbol period. 1 causes an internal pseudo-random number to be generated at the beginning of each symbol period.
`Seed`	This is the seed for the internal pseudo-random bit generator.

© 1994-2020 All Rights Reserved.

Figure 1-16 Constellation Map For Rectangualr_64_QAM

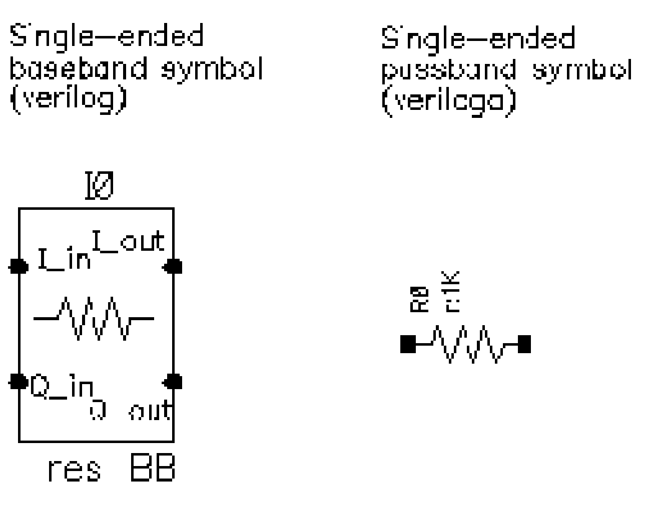




© 1994-2020
All Rights Reserved.

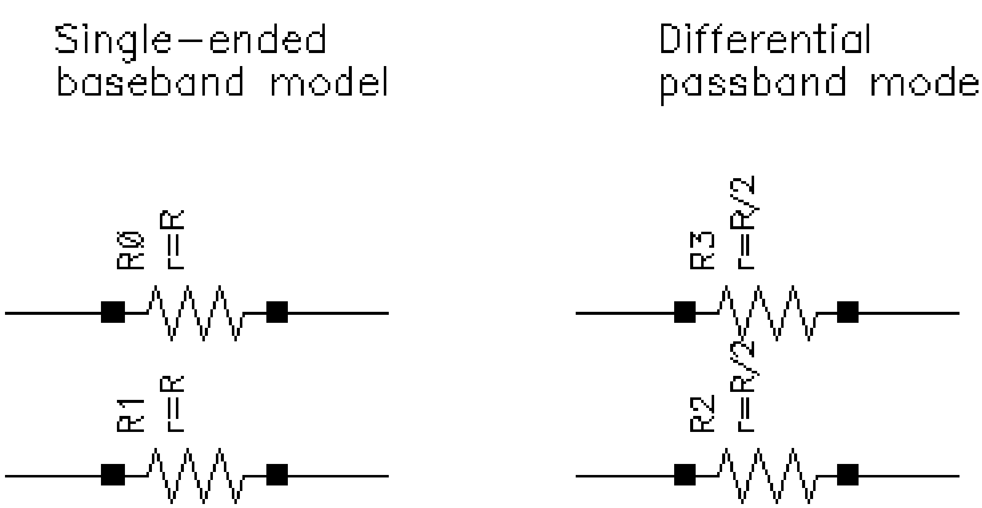
res_BB

Figure 1-17 Circuit



Besides the resistance, the baseband resistor model has a parameter for turning its thermal noise on or off. The baseband resistor is intended for use at a passband node because it's noise is doubled. (This was discussed in the section entitled "Relationship between baseband and passband noise"). Figure 1-18 shows the symbol, baseband, and passband models. The total noise in the differential passband resistor model equals the noise in one resistor of R Ohms.

Figure 1-18 Resistor Model



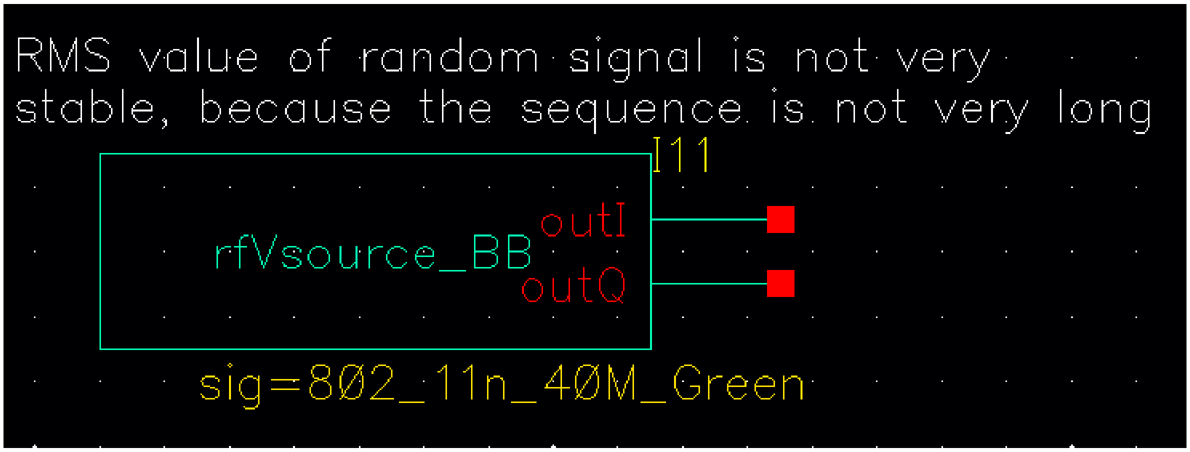
© 1994-2020
All Rights Reserved.

The parameters are:

`res`	
`noise_switch`	0 for noiseless, 1 for noisy.

© 1994-2020
All Rights Reserved.

rfVsourceBB



The parameters are:

`Sig_Standard`	The signal standard to be used. Choices are: 802.11a, 802.11G-ERP, 802.11n-20M-Mix 802.11n-20M-Legacy, 802.11n-20M-Green, 802.11n-40M-Mix, 802.11n-40m-Legacy, And 802.11n-40M-Green
`Power (dBm)`	RF Power in dBm
`Resistance (Ohms)`	System resistance in ohms.
`Filter`	Choices are none or erc (Raised Cosine)
`Roll off factor`	If erc is chosen for the filter, this parameter is displayed, and this is the rolloff factor of the raised cosine filter. The default is 0.23
`Band width (Hz)`	If erc is chosen for the filter, this parameter is displayed and this sets the bandwidth of the filter. If ACPR is to be measured, set this to twice the bandwidth of the main channel.

© 1994-2020
All Rights Reserved.

The rfVsourceBB is a baseband signal generator that provides an 802.11 modulated RF I and Q signals for use in a circuit. The bit sequence is a bit short, so this source should not be used to certify the spectral content of your power amplifier. It should be used for a quick idea only.

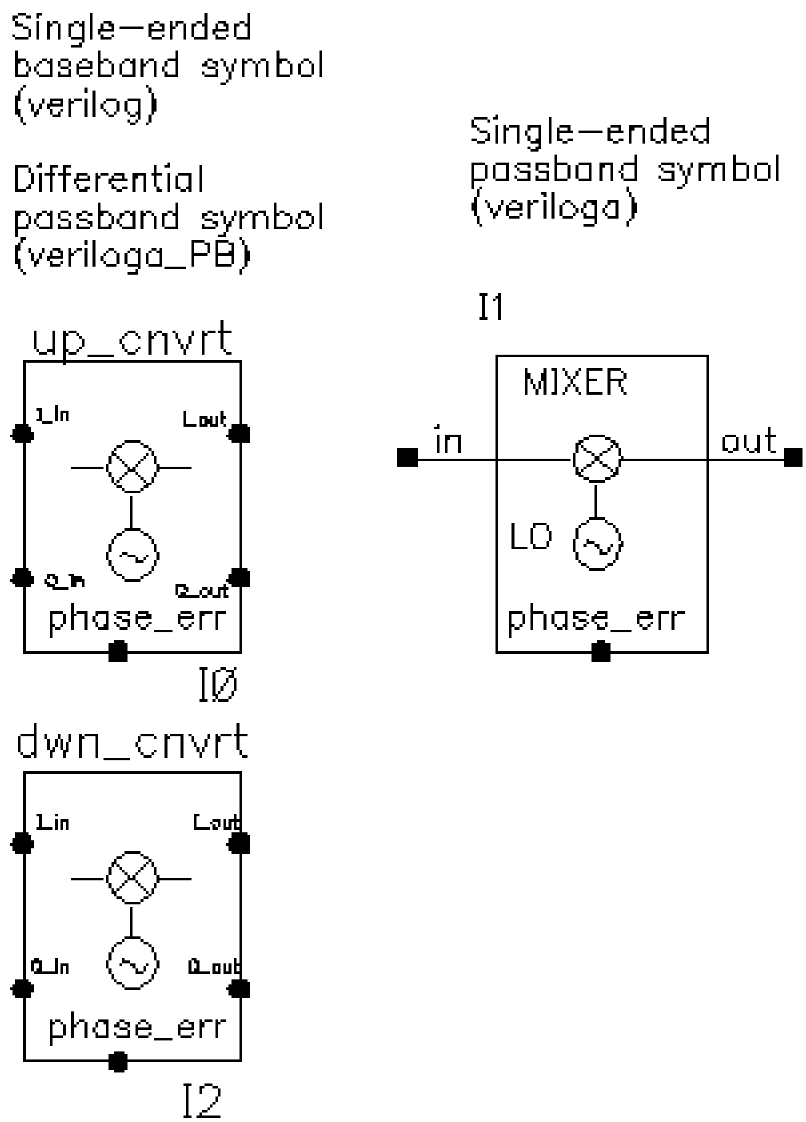
Signal Standard	Modulation Type	Data Length	Step Period	Symbol Start	FFT Size	Prefix size	Frame Duration	Packet Length	Source EVM
802.11a	QAM-64	819.1u	12.5n	4u	64	16	4u	173u	.000151%
802.11g-ERP	BPSK	1m	12.5n		64	16	4u	One packet	.002313%
802.11 20M-Mix	QAM-64	400u	8.33333333n	0	64	8	3.6u	147.6u	.011109%
802.11 20M-Legacy	QAM-64	352u	8.33333333n	0	64	8	3.6u	131.6u	.02672%
802.11 20M-Greenfield	QAM-64	376u	8.33333333n	0	64	8	3.6u	139.6u	.014321%
802.11 40M-Mix	QAM-64	400u	8.33333333n	0	128	16	3.6u	90u	.105442%
802.11 40M-Legacy	QAM-64	320u	8.33333333n	0	128	16	3.6u	74u	.25561%
802.11 40M-Greenfield	QAM-64	360u	8.33333333n	0	128	16	3.6u	82u	.10558%
802.11 ac	QAM-64	448u	4.16666667n		256	64	4u	One packet	.000065%

© 1994-2020
All Rights Reserved.

up_cnvrt

`(baseband = up_cnvrt)`

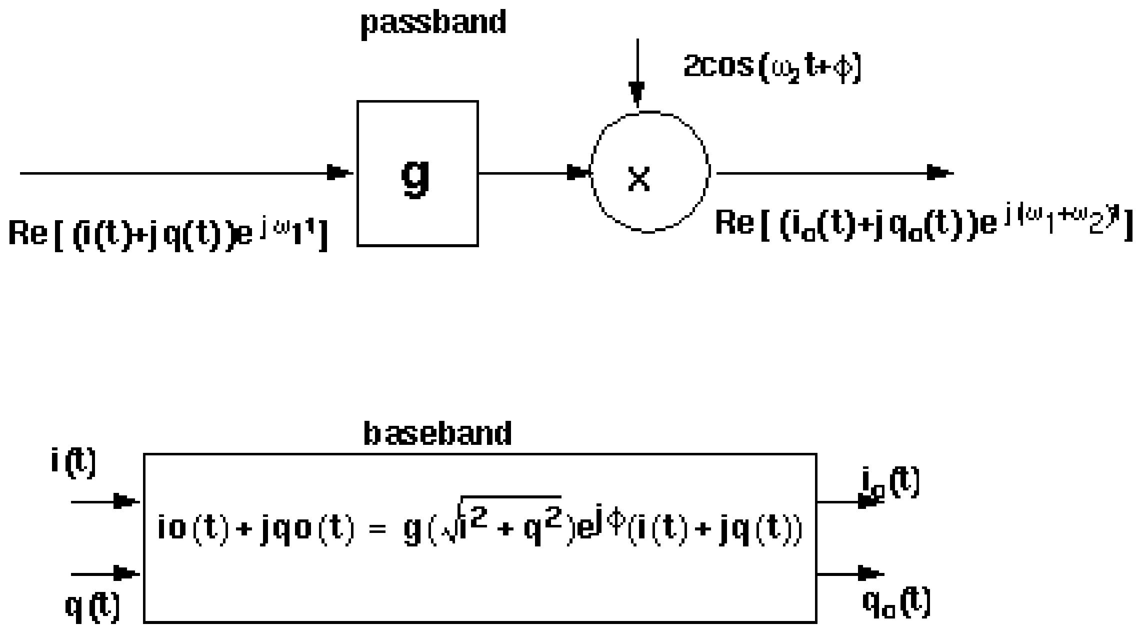
Figure 1-19 Baseband and Passband Mixer Models



`up_cnvrt` model is a baseband equivalent model of a mixer used to convert from IF to RF. There are some minor differences in the baseband models that depend on whether conversion is up or down. Figures 1-20 shows what the model does.

© 1994-2020 All Rights Reserved.

Figure 1-20 Calculations for up_cnvrt Mixer



The parameters are:

`cmp`	Input power pt for phase point [dBm].
`cw`	{1,0,-1} for {cw, none, ccw}. Defines the direction of the phase shift. 1 for clockwise, 0 for no phase shift, -1 for counter clockwise.
`gain`	Available power gain [dB].
`IP3`	Input referenced IP3 [dbm].
`nf`	Noise figure.
`pscp`	|radians|@cmp. Defines the absolute value of the output phase shift at the 1dB compression point for power amplifiers. This is the phase shift at an arbitrary output power level for some models.
`psinf`	|radians|@ big input. Defines the absolute value of the output phase shift as input power goes to infinity (if it could go to infinity).
`rin`	Input resistance.
`rout`	Output resistance.

© 1994-2020
All Rights Reserved.

`shp`	AM/PM sharpness. Determines how fast the phase shift occurs with increasing input power. A larger number delays the shift but makes the shift rise faster as a function of input signal level.

© 1994-2020 All Rights Reserved.

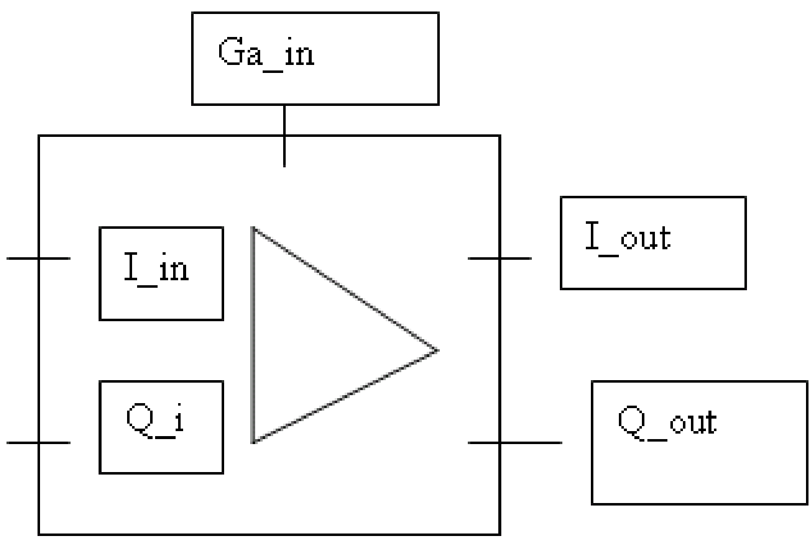
VGA_BB

(Variable Gain Amplifier Model)

`(baseband = VGA_BB)`

Only the Baseband view is available.

Figure 1-21 Variable Gain Amplifier Model



The parameters are:

`cpdb`	db compression point [dbm], referred to the output.
`cw`	{1,0,-1} for {cw, none, ccw}. Defines the direction of the phase shift. 1 for clockwise, 0 for no phase shift, -1 for counter clockwise.
`gpv`	Voltage gain per volt on the G_in pin.
`pscp`	|radians|@cmp. Defines the absolute value of the output phase shift at the 1dB compression point for power amplifiers. This is the phase shift at an arbitrary output power level for some models.
`psinf`	|radians|@ big input. Defines the absolute value of the output phase shift as input power goes to infinity (if it could go to infinity).
`rin`	Input resistance.
`rout`	Output resistance.

© 1994-2020
All Rights Reserved.

`shp`	AM/PM sharpness. Determines how fast the phase shift occurs with increasing input power. A larger number delays the shift but makes the shift rise faster as a function of input signal level.

© 1994-2020 All Rights Reserved.

Butterworth_filters Category

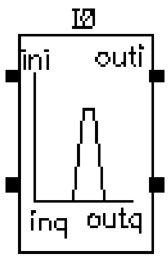
The cells in the `Butterworth_filters` category are:

- BB_butterworth_bp
- BB_butterworth_bs
- BB_butterworth_hp
- BB_butterworth_lp
- butterworth_bp
- butterworth_bs
- butterworth_hp
- butterworth_lp

© 1994-2020 All Rights Reserved.

BB_butterworth_bp

Figure 1-22 BB_butterworth_bp



For information about the filter parameters, see "BB_butterworth and BB_chebyshev Filter Parameters" on page 67.

BB_butterworth and BB_chebyshev Filter Parameters

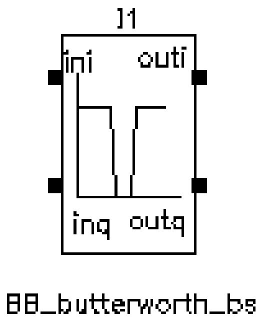
The parameters for the BB_butterworth and BB_chebyshev filters are:

`bw`	Relative bandwidth for bandpass or bandstop filter [Hz].
`f0`	Center frequency for bandpass or bandstop filter [Hz].
`fc`	Filter cutoff frequency for lowpass and highpass filter [Hz].
`fcr`	Carrier frequency.
`loss`	Filter insertion loss [dB].
`N`	Filter order (>=2). Must be defined as ‘define N.
`r1`	Reference impedance at port 1 [ohm].
`r2`	Reference impedance at port 2 [ohm].

© 1994-2020 All Rights Reserved.

BB_butterworth_bs

Figure 1-23 BB_butterworth_bs

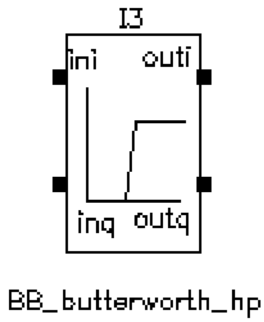


For information about the filter parameters, see “BB_butterworth and BB_chebyshev Filter Parameters” on page 67.

© 1994-2020
All Rights Reserved.

BB_butterworth_hp

Figure 1-24 BB_butterworth_hp

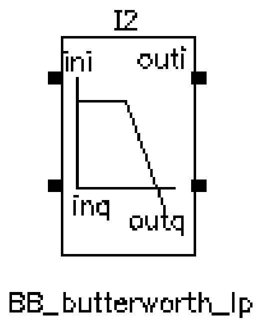


For information about the filter parameters, see "BB_butterworth and BB_chebyshev Filter Parameters" on page 67.

© 1994-2020
All Rights Reserved.

BB_butterworth_lp

Figure 1-25 BB_butterworth_lp

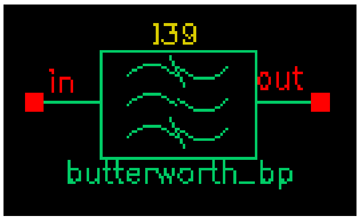


For information about the filter parameters, see “BB_butterworth and BB_chebyshev Filter Parameters” on page 67.

© 1994-2020
All Rights Reserved.

butterworth_bp

Figure 1-26 butterworth_bp



The parameters are:

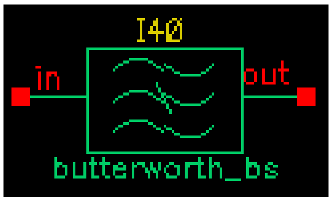
`alpha`	Filter attenuation at cutoff [dB].
`bw`	Relative bandwidth for bandpass or bandstop filter [Hz].
`f0`	Center frequency for bandpass or bandstop filter [Hz].
`fc`	Filter cutoff frequency for lowpass and highpass filter [Hz].
`loss`	Filter insertion loss [dB].
`N`	Filter order (>=2). Must be defined as ‘define N.
`r1`	Reference impedance at port 1 [Ohm].
`r2`	Reference impedance at port 2 [Ohm].

For more information, see “Butterworth and Chebyshev Filter Supporting Information” on page 75.

© 1994-2020 All Rights Reserved.

butterworth_bs

Figure 1-27 butterworth_bs



The parameters are:

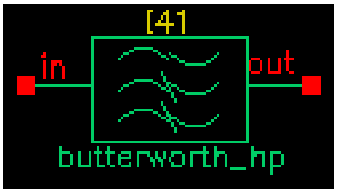
`alpha`	Filter attenuation at cutoff [dB].
`bw`	Relative bandwidth for bandpass or bandstop filter [Hz].
`f0`	Center frequency for bandpass or bandstop filter [Hz].
`fc`	Filter cutoff frequency for lowpass and highpass filter [Hz].
`loss`	Filter insertion loss [dB].
`N`	Filter order (>=2). Must be defined as ‘define N.
`r1`	Reference impedance at port 1 [Ohm].
`r2`	Reference impedance at port 2 [Ohm].

For more information, see “Butterworth and Chebyshev Filter Supporting Information” on page 75.

© 1994-2020 All Rights Reserved.

butterworth_hp

Figure 1-28 butterworth_hp



The parameters are:

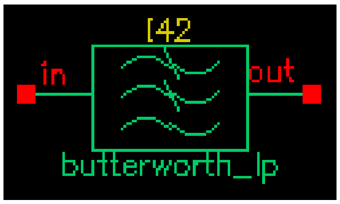
`alpha`	Filter attenuation at cutoff [dB].
`bw`	Relative bandwidth for bandpass or bandstop filter [Hz].
`f0`	Center frequency for bandpass or bandstop filter [Hz].
`fc`	Filter cutoff frequency for lowpass and highpass filter [Hz].
`loss`	Filter insertion loss [dB].
`N`	Filter order (>=2). Must be defined as ‘define N.
`r1`	Reference impedance at port 1 [Ohm].
`r2`	Reference impedance at port 2 [Ohm].

For more information, see “Butterworth and Chebyshev Filter Supporting Information” on page 75.

© 1994-2020 All Rights Reserved.

butterworth_lp

Figure 1-29 butterworth_lp



The parameters are:

`alpha`	Filter attenuation at cutoff [dB].
`bw`	Relative bandwidth for bandpass or bandstop filter [Hz].
`f0`	Center frequency for bandpass or bandstop filter [Hz].
`fc`	Filter cutoff frequency for lowpass and highpass filter [Hz].
`loss`	Filter insertion loss [dB].
`N`	Filter order (>=2). Must be defined as ‘define N.
`r1`	Reference impedance at port 1 [Ohm].
`r2`	Reference impedance at port 2 [Ohm].

For more information, see “Butterworth and Chebyshev Filter Supporting Information” on page 75.

© 1994-2020 All Rights Reserved.

Butterworth and Chebyshev Filter Supporting Information

Filter properties are specified in the frequency domain, but it is not easy for Spectre RF to process frequency-domain data. Spectre RF simulation requires a large signal, time-domain model to simulate filter behavior.

As part of the RF AHDL library, filters are implemented using a network synthesis technique which consists of the following two steps:

1. Calculate the normalized low-pass filter prototype, which consists of serial inductors and parallel capacitors
2. Perform frequency transformation and scaling to synthesize the frequency responses of the filter type

The synthesized model contains many inductors and capacitors. They are implemented using the integral and differential functions of the Verilog-A language. Insertion loss is added using the S-parameter network technique. This network essentially dampens the signal flow by the specified insertion loss value.

In the current implementation of the Verilog-A language, the order and internal states of the filter cannot be dynamically allocated. You must use the `'define` directive in the Verilog-A source code to specify the order. Use S-parameters to test the filters because S-parameters capture the input/output impedance matching.

For example, the Butterworth bandpass filter, `butterworth_bp`, has the following module declaration:

```
module butterworth_bp(t1, t2);
    inout t1, t2;
    electrical in, out;
    parameter real r1 = 50 from (0:inf);
    parameter real r2 = 50 from (0:inf);
    parameter real f0 = 1e9 from (0:inf);
    parameter real bw = 0.10 from (0:0.5);
    parameter real fc = 1e9 from (0:inf);
    parameter real loss = 0 from [0:inf);
```

where `t1` and `t2` are the input and output nodes, respectively.

The parameters are:

`bw`	Relative frequency for bandpass or bandstop filter [Hz].
`f0`	Center frequency for bandpass or bandstop filter [Hz].
`fc`	Corner frequency (3 dB point) for low-pass and high-pass filter [Hz].

© 1994-2020 All Rights Reserved.

`loss`	Insertion loss [dB].
`r1`	Input impedance [Ω].
`r2`	Output impedance [Ω].

Figure 1-30 is the simple schematic used to test the filter. Two ports are used to obtain the S-parameters.

Figure 1-30 Schematic for Testing Filter Models

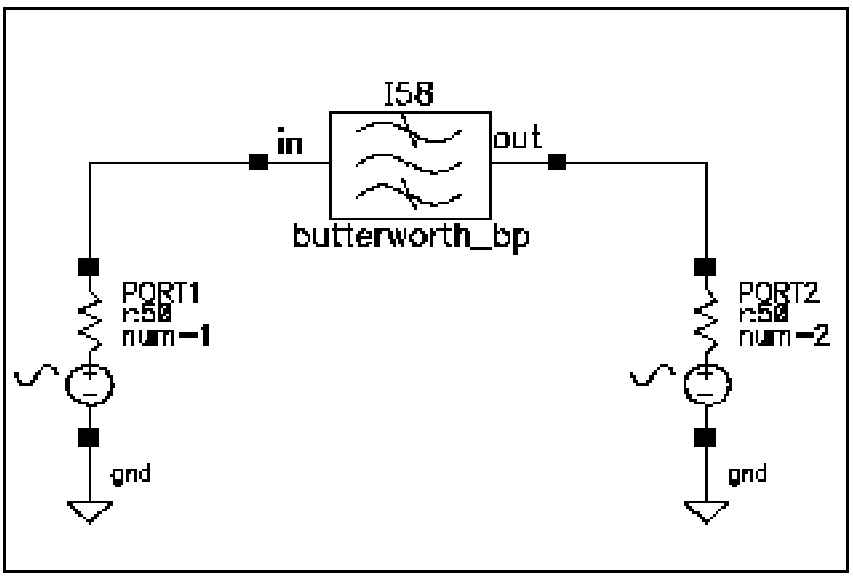
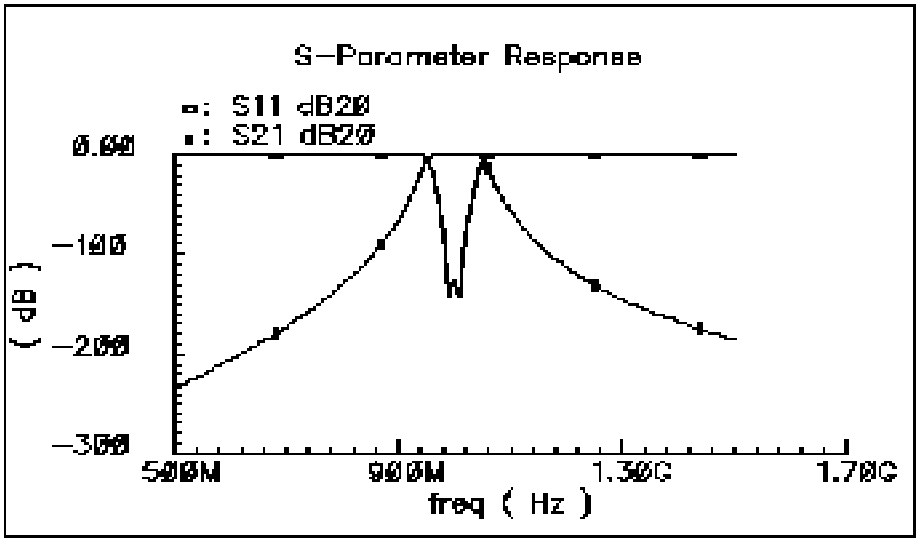


Figure 1-31 shows the calculated S-parameters of this Butterworth bandpass filter, which has a center frequency of 1 GHz and a relative bandwidth of 10 percent. The order of this specific filter is 10.

© 1994-2020 All Rights Reserved.

Figure 1-31 S-Parameters of a Butterworth Filter



© 1994-2020
All Rights Reserved.

Chebyshev_filters Category

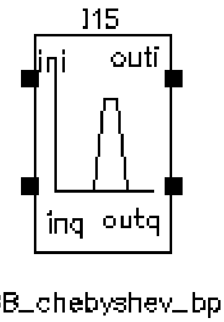
The cells in the `Chebyshev_filters` category are:

- BB_chebyshev_bp
- BB_chebyshev_bs
- BB_chebyshev_hp
- BB_chebyshev_lp
- chebyshev_bp
- chebyshev_bs
- chebyshev_hp
- chebyshev_lp

© 1994-2020 All Rights Reserved.

BB_chebyshev_bp

Figure 1-32 BB_chebyshev_bp

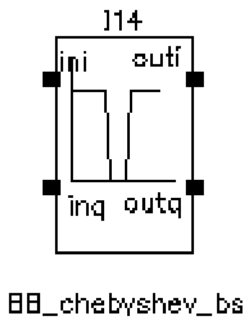


For information about the filter parameters, see "BB_butterworth and BB_chebyshev Filter Parameters" on page 67.

© 1994-2020
All Rights Reserved.

BB_chebyshev_bs

Figure 1-33 BB_chebyshev_bs

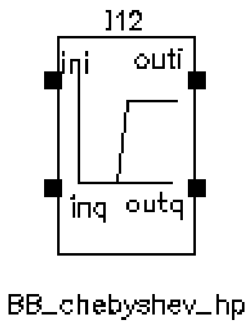


For information about the filter parameters, see "BB_butterworth and BB_chebyshev Filter Parameters" on page 67.

© 1994-2020
All Rights Reserved.

BB_chebyshev_hp

Figure 1-34 BB_chebyshev_hp

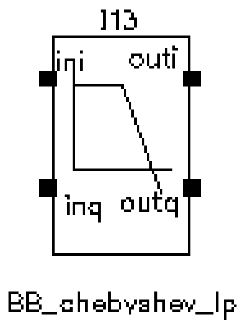


For information about the filter parameters, see “BB_butterworth and BB_chebyshev Filter Parameters” on page 67.

© 1994-2020
All Rights Reserved.

BB_chebyshev_lp

Figure 1-35 BB_chebyshev_lp

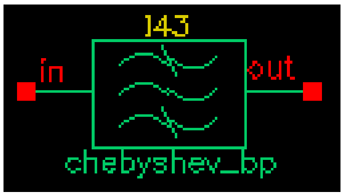


For information about the filter parameters, see “BB_butterworth and BB_chebyshev Filter Parameters” on page 67.

© 1994-2020
All Rights Reserved.

chebyshev_bp

Figure 1-36 chebyshev_bp



The parameters are:

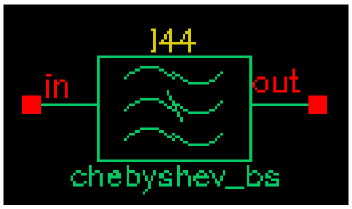
`alpha`	Filter attenuation at cutoff [dB].
`bw`	Relative bandwidth for bandpass or bandstop filter [Hz].
`f0`	Center frequency for bandpass or bandstop filter [Hz].
`fc`	Filter cutoff frequency for lowpass and highpass filter [Hz].
`loss`	Filter insertion loss [dB].
`N`	Filter order (>=2). Must be defined as ‘define N.
`r1`	Reference impedance at port 1 [Ohm].
`r2`	Reference impedance at port 2 [Ohm].

For more information, see “Butterworth and Chebyshev Filter Supporting Information” on page 75.

© 1994-2020 All Rights Reserved.

chebyshev_bs

Figure 1-37 chebyshev_bs



The parameters are:

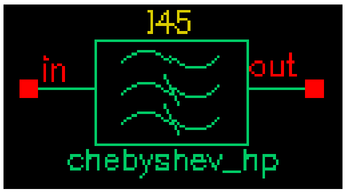
`alpha`	Filter attenuation at cutoff [dB].
`bw`	Relative bandwidth for bandpass or bandstop filter [Hz].
`f0`	Center frequency for bandpass or bandstop filter [Hz].
`fc`	Filter cutoff frequency for lowpass and highpass filter [Hz].
`loss`	Filter insertion loss [dB].
`N`	Filter order (>=2). Must be defined as ‘define N.
`r1`	Reference impedance at port 1 [Ohm].
`r2`	Reference impedance at port 2 [Ohm].

For more information, see “Butterworth and Chebyshev Filter Supporting Information” on page 75.

© 1994-2020 All Rights Reserved.

chebyshev_hp

Figure 1-38 chebyshev_hp



The parameters are:

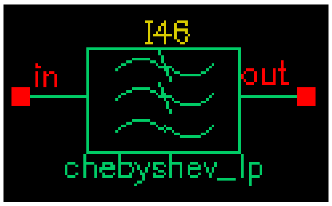
`alpha`	Filter attenuation at cutoff [dB].
`bw`	Relative bandwidth for bandpass or bandstop filter [Hz].
`f0`	Center frequency for bandpass or bandstop filter [Hz].
`fc`	Filter cutoff frequency for lowpass and highpass filter [Hz].
`loss`	Filter insertion loss [dB].
`N`	Filter order (>=2). Must be defined as ‘define N.
`r1`	Reference impedance at port 1 [Ohm].
`r2`	Reference impedance at port 2 [Ohm].

For more information, see “Butterworth and Chebyshev Filter Supporting Information” on page 75.

© 1994-2020 All Rights Reserved.

chebyshev_lp

Figure 1-39 chebyshev_lp



The parameters are:

`alpha`	Filter attenuation at cutoff [dB].
`bw`	Relative bandwidth for bandpass or bandstop filter [Hz].
`f0`	Center frequency for bandpass or bandstop filter [Hz].
`fc`	Filter cutoff frequency for lowpass and highpass filter [Hz].
`loss`	Filter insertion loss [dB].
`N`	Filter order (>=2). Must be defined as ‘define N.
`r1`	Reference impedance at port 1 [Ohm].
`r2`	Reference impedance at port 2 [Ohm].

For more information, see “Butterworth and Chebyshev Filter Supporting Information” on page 75.

© 1994-2020 All Rights Reserved.

Measurements Category

The `measurement` category contains elements used to facilitate measurements and diagnostics. Elements in the `measurement` category can be used by both RF system designers and RF circuit designers.

This section also explains how to change the FIR filters inside the baseband signal generators.

Note: All of the baseband signal sources generate digitally filtered signals. The baseband sources do not work with Spectre RF because the digital filters have hidden states.

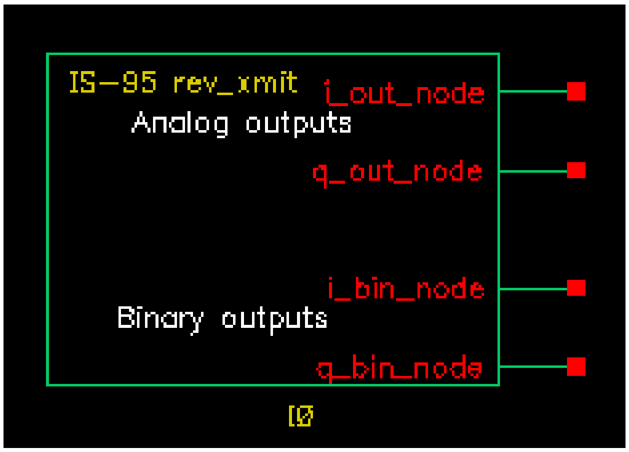
The `measurement` category contains the following elements, discussed in the sections that follow.

- CDMA_reverse_xmit
- comms_instr
- eye_diagram_generator
- gmsk
- GSM_xmtr
- instr_term
- offset_comms_instr
- pi_over4_dqpsk
- polar_rect
- rect_polar

© 1994-2020
All Rights Reserved.

CDMA_reverse_xmit

(CDMA Signal Source)



The CDMA signal source (`CDMA_reverse_xmit`) generates a reverse-link (handset-to-base-station) IS-95 signal with the following characteristics

modulation	Offset QPSK.
symbol rate	1.2288 megasymbols per second.
sample rate	4.9152 megasamples per second.

Two separate 16-bit pseudo-noise generators generate the *I* and *Q* spreading sequences operating at the sample rate.

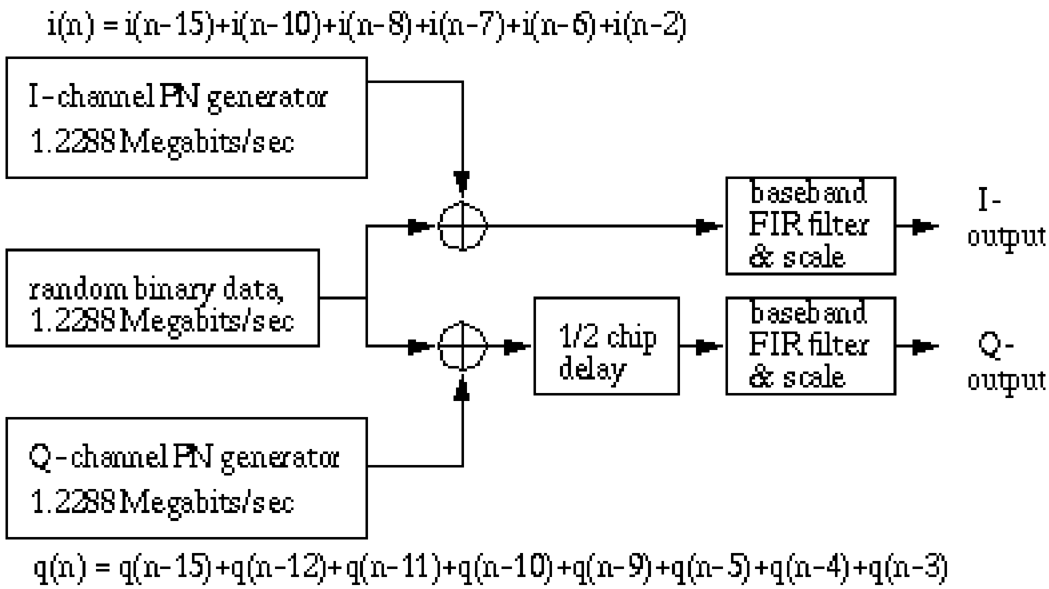
The CDMA source

- Generates a random bit at the symbol rate
- Oversamples it by a factor of 4
- Spreads the bit with the *I* and *Q* spreading sequences
- Filters each sequence with a 48-tap FIR filter. The filter coefficients are the impulse response of a raised cosine filter.
- Generates a reverse-link (handset-to-base-station) IS-95 signal. The modulation is offset QPSK with a symbol rate of 1.2288 Mega-symbols per second and a sample rate of 4.9152 Megasamples per second. Two separate 16-bit pseudo-noise generators generate the *I* and *Q* spreading sequences operating at the sample rate.

© 1994-2020 All Rights Reserved.

Figure 1-40 shows a block diagram of the signal generator.

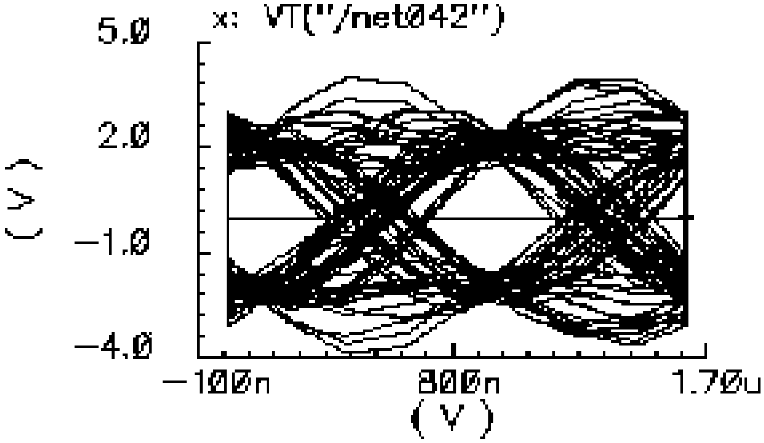
Figure 1-40 CDMA Baseband Test Signal Generator

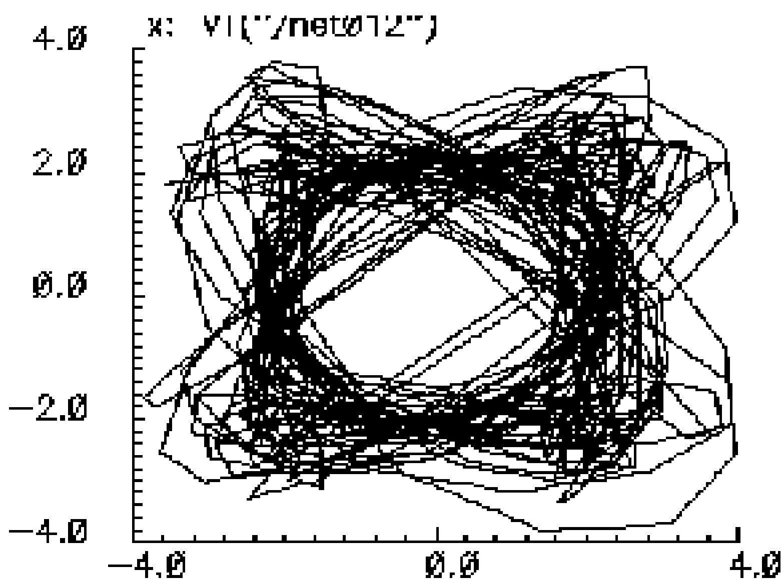


The eye-diagram generator (`eye_diagram_generator`) created the eye-diagram and trajectory. Figure 1-41 on page 90 shows the eye-diagram of one of the outputs and the trajectory of both outputs.

© 1994-2020
All Rights Reserved.

Figure 1-41 Eye Diagram and CDMA Trajectory





CDMA Signal Source Instance Parameters

The `amplitude` parameter sets the amplitude of the unfiltered signals. An amplitude of 1 means that each FIR filter is driven by 1 volt impulses. If you change the internal variable `IMPULSE_PULSE` to 2, the filters are driven by 1 volt pulses of four samples duration.

The `seed` parameter changes the seed for the random number generator.

All Rights Reserved.

CDMA Signal Source Outputs

The CDMA signal generator creates four output signals:

`i_bin_node`	The *I* unfiltered binary output.
`i_out_node`	The filtered *I* output.
`q_bin_node`	The *Q* unfiltered binary output.
`q_out_node`	The *Q* filtered output.

Changing the FIR Filter in a CDMA Signal Source

You cannot change the FIR filter, such as the tap length and tap coefficients, directly from the instance. However, you can do so using the Modelwriter as described in “Modifying the BB Signal Generators Using Modelwriter” on page 294.

CDMA Signal Source Output Transitions

The filtered outputs slew linearly from one value to the next because the rise and fall times in the transition statements equal one period. To make the outputs take abrupt steps, copy the module to your library and change the rise and fall times in the last transition statements.

© 1994-2020 All Rights Reserved.

comms_instr

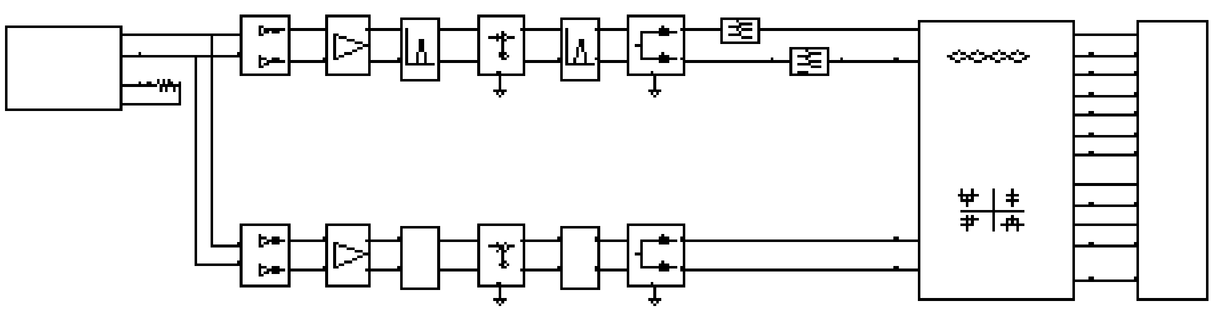
(Instrumentation Block)

The `comms_instr` block generates waveforms that can be used to create eye-diagrams, eye-diagram statistics, scatter plots, and rms error-vector-magnitudes.

For information about the parameters of the block, see “Instrumentation Block Parameters” on page 93. For information about the outputs of the block, see “Instrumentation Block Outputs” on page 94. For information about the related `offset_comms_instr` block, see “(Instrumentation Block)” on page 103. For information about the related `instr_term` block, see “instr_term” on page 102.

Figure 1-42 on page 92 shows how the `offset_comms_instr` and `instr_term` blocks should be used. The `comms_instr` block is used similarly. The circuit consists of two branches driven from a single baseband signal generator. The top branch is the non-ideal receiver model, the bottom branch is an ideal version of the top branch. The ideal version is as ideal as you like. The ideal branch computes ideal symbol locations in the complex plane. The instrumentation block compares ideal and non-ideal symbols to compute the error-vector-magnitude.

Figure 1-42 EVM setup



© 1994-2020
All Rights Reserved.

Instrumentation Block Parameters

Parameter	Meaning
`I_del`	**I-sampling delay.** This parameter sets the phase of the symbol sampler. It is referenced to the eye-diagram of the I-output. Estimate the optimal delay by doing one simulation just to get the eye-diagram. The optimal delay is the time from the leftmost part of the eye-diagram, which should be zero, to the time at which they eye is widest.
`I_noise`, `Q_noise`	**I-noise and Q-noise(volts^2).** These parameters set the variance of Gaussian random variables which can be added to the received symbols before anything is computed or plotted.
`max_voltage`, `min_voltage`, `num_of_bins`	**Max, min eye-diagram volts, and number of hstgm bins.** These parameters are used to compute the bins which define the eye-diagram histogram. The bin width equals (max voltage-min voltage)/(number of bins). The histogram shows the distribution of the I_in voltage at the sampling instant.
`measurement_delay`	**Statistics start time.** This parameter delays the start of any statistical computations. The purpose is to exclude start-up transients from the statistics.
`number_of_symbols`	**Number of symbols.** This is the number of symbols to sweep in the eye-diagram. Sweeping two symbols ensures that you see at least one continuous eye, if it is open.
`rin`	**Input resistance.** This parameter is the input resistance of the input terminals of the instrumentation block.
`symbol_rate`	**Symbols per second.** This parameter is necessary for generating the sawtooth that is used as the x-axis to generate eye-diagrams. It also determines the rate at which the input waveforms are sampled.

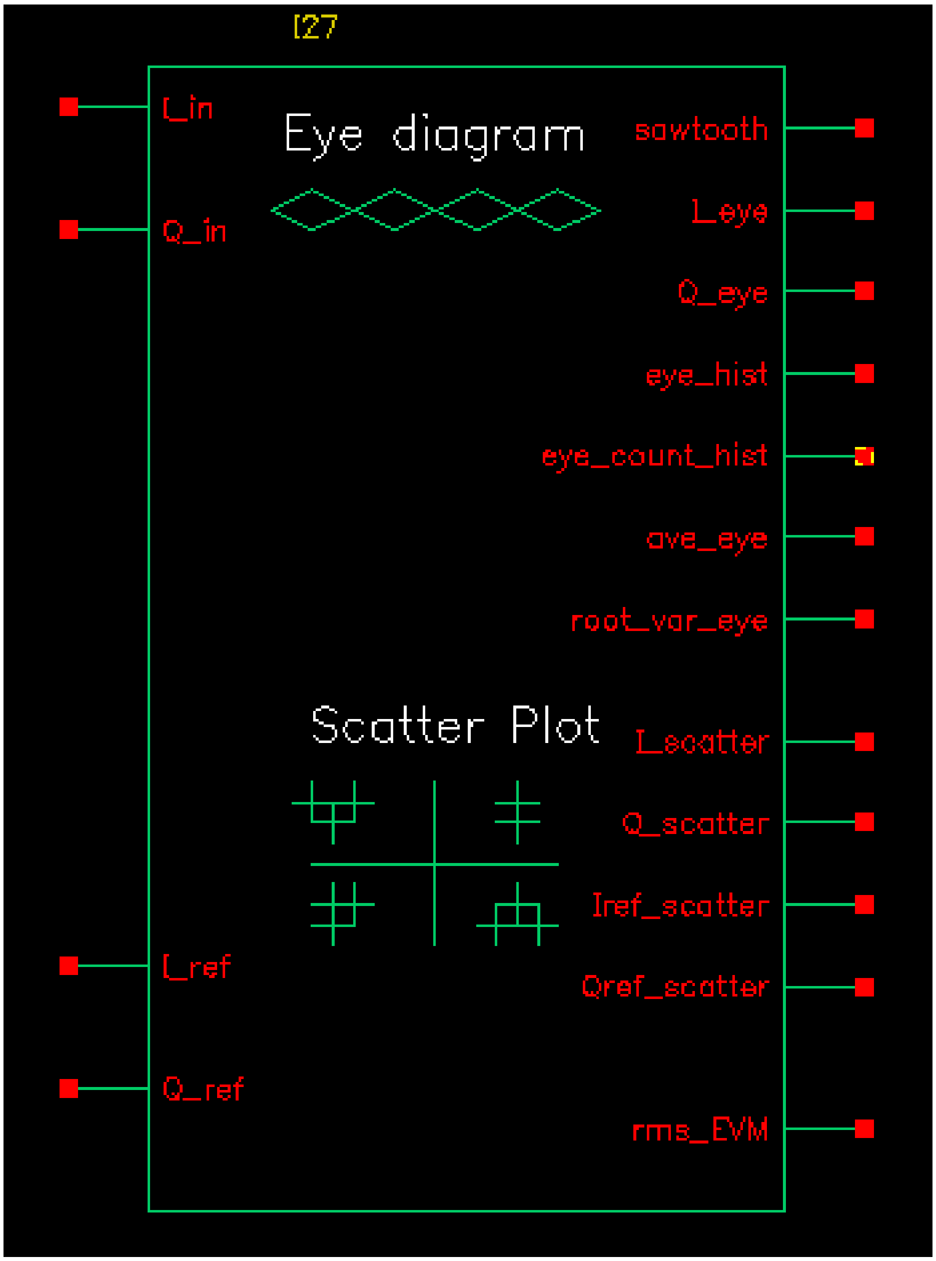
© 1994-2020
All Rights Reserved.

Instrumentation Block Outputs

Output	Meaning
`ave_eye`, `root_var_eye`	**Eye-diagram statistics.** The ave_eye output is the average absolute value of the I-input signal at the sampling instant. The root_var_eye output is the square root of the variance of the absolute value of the I-input voltage at the sampling instant. The voltages at these output pins represent running estimates of the associated statistics.
`eye_hist`, `eye_count_hist`	**Histograms.** You can only generate a histogram of the I-input signal. The histogram shows the distribution of the I-input voltage at the sampling instant. To generate a histogram, plot the eye_hist and eye_count_hist outputs in the same waveform display window. Change the x-axis to be the eye-hist signal then change the plot to *bar*.
`I_scatter`,`Q_scatter`, `Iref_scatter`, `Qref_scatter`	**Scatter plots.** A scatter plot is the I-input and Q-input samples plotted against each other. The scatter plot shows the locations of the received symbols. To generate a scatter plot, plot I_scatter and Q_scatter in the same waveform display window then change the x-axis to be the I_scatter signal. Finally, change the plot to plot data points only. A scatter plot of the reference model can be generated similarly by replacing I_scatter and Q_scatter with Iref_scatter and Qref_scatter.
`rms_EVM`	**RMS Error Vector Magnitude.** The rms EVM is defined as the square root of the sum of the squares of the vectorial differences between the ideal and non-ideal received symbols, normalized to the rms value of the magnitude of the ideal received symbols. The output voltage at this pin is represents a running calculation of the rms EVM. The normalized EVM is in percent.
`sawtooth`, `I_eye`, `Q_eye`	**Eye-diagrams.** To generate an eye-diagram of the I-input signal, plot the sawtooth and I-eye outputs in one waveform display tool. Change the x-axis to be the sawtooth. This is done through the x-axis menu in the waveform display tool. The procedure for generating an eye-diagram of the Q-output is the same except you use the Q-eye output.

© 1994-2020
All Rights Reserved.

eye_diagram_generator



The eye-diagram generator creates eye-diagrams and trajectories for the baseband signal generators. For more information, see "Eye-Diagram Generator Input" on page 96 and "Eye-Diagram Generator Outputs" on page 96.

Eye-Diagram Generator Input

The input to the eye-diagram generator is the *I* or *Q* output of one of the baseband signal generators.

Eye-Diagram Generator Outputs

The eye-diagram generator has two outputs labeled `y-axis` and `x-axis`. The eye-diagram is generated by plotting the `y-axis` output against the `x-axis` output.

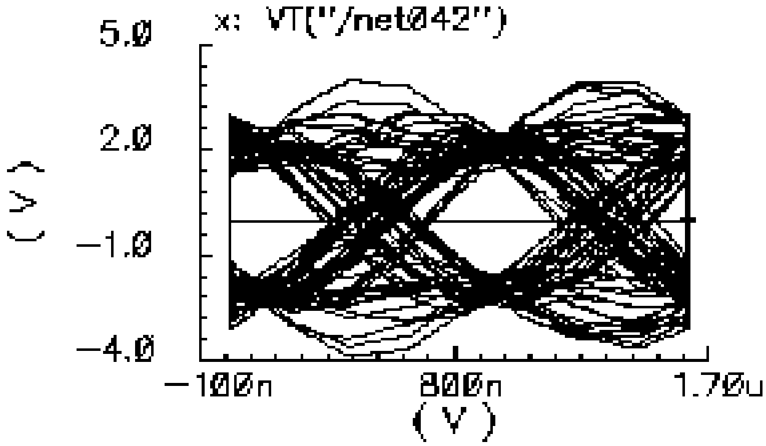
The eye-diagram generator does not work with Envelope analysis to generate similar plots. This is because the Envelope harmonic time analysis is generated by a post-processing step and the eye-diagram generator works during simulation.

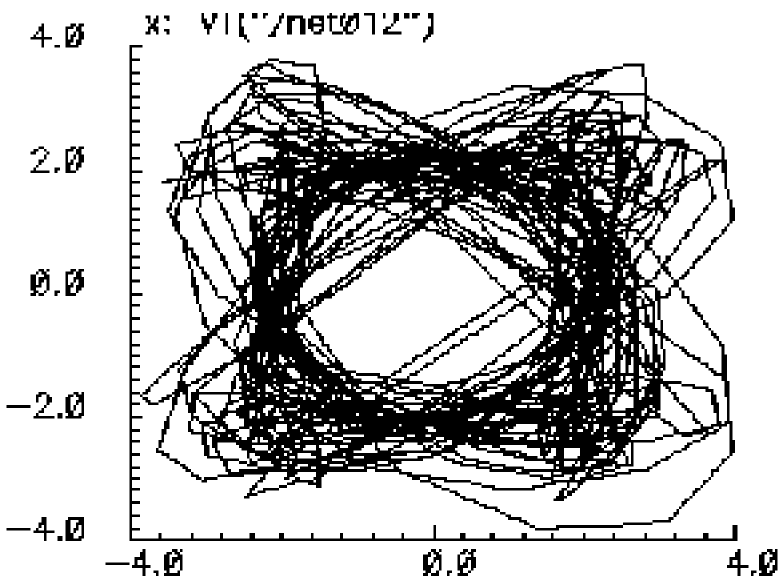
Figure 1-43 shows an eye-diagram of one of the outputs and the trajectory of both outputs for the CDMA baseband signal generator.

© 1994-2020

All Rights Reserved.

Figure 1-43 Example Eye Diagram and CDMA Trajectory





© 1994-2020 All Rights Reserved.

gmsk

GMSK (Gaussian minimum shift keying) is a simple but efficient approach to digital modulation that provides the properties of narrow-band techniques, sharp cutoffs in frequency, lower overshoot impulse response, and preservation of the filter output pulse area. These qualities result in low phase distortion and make GMSK suitable for coherent demodulation. The GMSK approach is used in the Global System for Mobile Communication (GSM).

In release IC 6.1.2 and later, Cadence provides a Verilog-A module for simulating GMSK behavior. The module is located in `rfLib`.

Figure 1-44 GMSK symbol

The `inbit` input supports the use of external random generators. Usually `inbit` is disabled by specifying `enable_input` = 0.

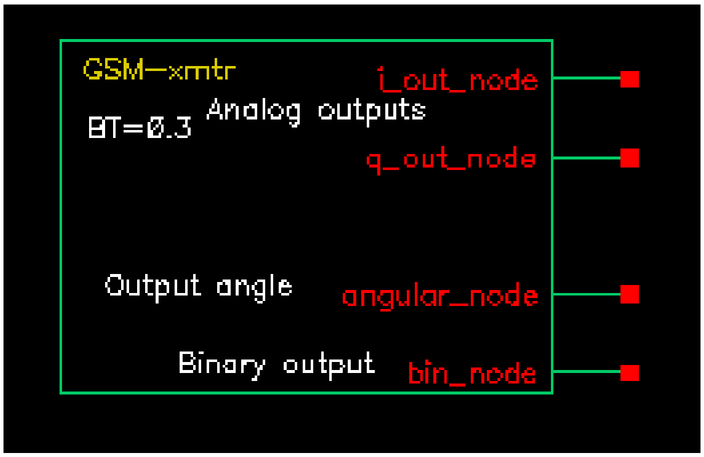
The parameters of the instance are listed in the table below:

Name	**Meaning**	**Type**	**Default Value**	**Range**
`BT`	BT product.	real	0.3	
`Period`	Input bits period.	real	6.0/1625000	
`Phase_offset`	Initial phase, normalized by p_i	real	0	
`Pulse_length`	Pulse length of Gaussian Filter	integer	3	
`Samples`	Number of samples in one T	integer	16	
`initial_sym`	Prehistory symbols used before simulation	integer	{0, 0, 0}	1 or -1
`seed`	Used for internal random generator when input is disabled	type	21	
`Enable_input`	1 if using external inputs (NRZ code)	integer	1	0 or 1

© 1994-2020
All Rights Reserved.

GSM_xmtr

(GSM Signal Source)



The GSM source generates a signal conforming to the GSM standard. The modulation is GMSK and the data is generated in frames of 3 fixed start bits, 142 random data bits, 3 fixed stop bits, and 8.25 fixed guard bits. (The embedded deterministic pattern and quarter of a bit is necessary to produce the correct spectrum.) The bit rate is 270833.333 bits per second and the sample rate is four times that.

The FIR filter is a Gaussian filter implemented with 32 taps.

Figure 1-45 shows a block diagram of the signal source.

Figure 1-45 GSM Baseband Signal Generator

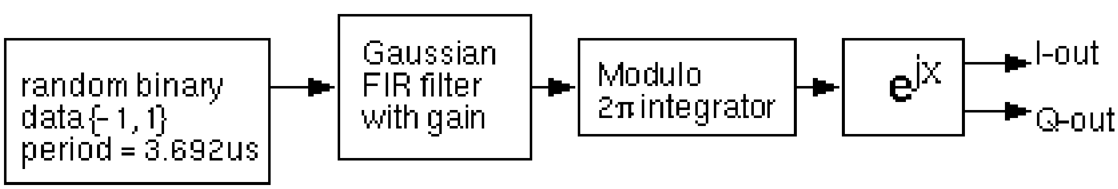
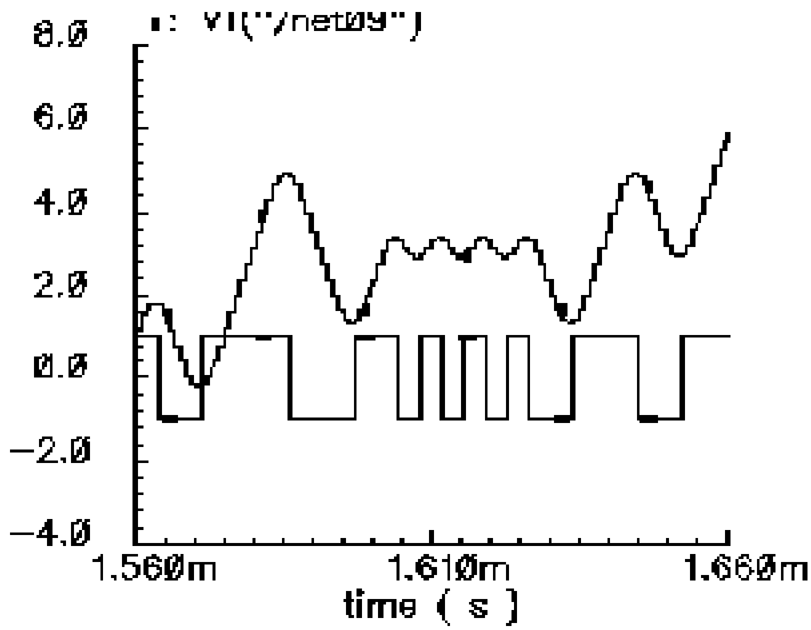


Figure 1-46 shows the binary data stream and the corresponding angle.

© 1994-2020
All Rights Reserved.

Figure 1-46 GSM Binary Data and Resulting Phase



GSM Signal Source Instance Parameters

The `amplitude` parameter sets the amplitude of the unfiltered signals. An amplitude of 1 means that each FIR filter is driven by 1-volt impulses. If you change the internal variable `IMPULSE_PULSE` to `2`, the filters are driven by 1-volt pulses of four samples duration.

The `seed` parameter changes the seed for the random number generator.

GSM Signal Source Outputs

The generator creates four output signals:

`angular_node`	The output signal.
`i_out_node`	The phase, multiplied by the amplitude.
`bin_node`	The bit stream being transmitted.
`q_out_node`	The phase multiplied by the amplitude.

© 1994-2020 All Rights Reserved.

Changing the FIR Filter in a GSM Signal Source

You cannot directly change the FIR filter, such as the tap length and tap coefficients, from the instance. However, you can make changes using the Modelwriter as described in "Modifying the BB Signal Generators Using Modelwriter" on page 294.

GSM Signal Source Output Transitions

The filtered outputs slew linearly from one value to the next because the rise and fall times in the transition statements equal one period. To make the outputs take abrupt steps, copy the module to your library and change the rise and fall times in the last transition statements.

© 1994-2020 All Rights Reserved.

instr_term

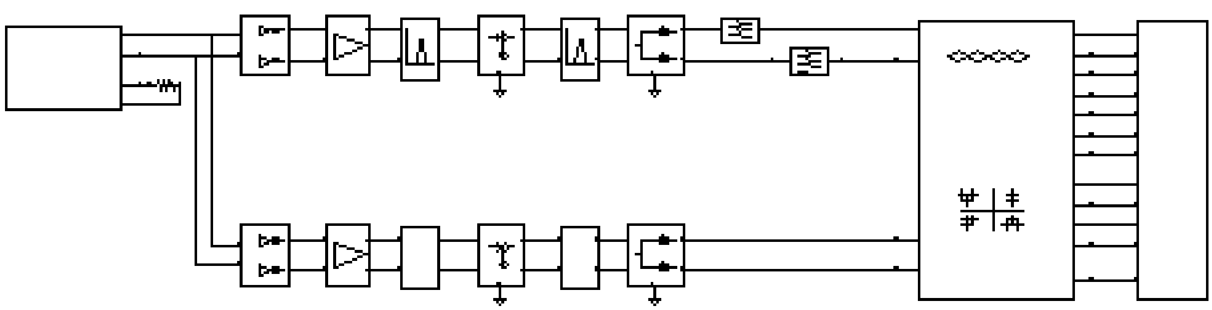
(Terminating Block)

The `instr_term` block simply loads all instrumentation output pins with 50 Ohms. The `instr_block` keeps the schematic editor from complaining about unconnected pins, nothing more.

For information about the parameters of the block, see “Instrumentation Block Parameters” on page 93. For information about the outputs of the block, see “Instrumentation Block Outputs” on page 94. For information about the related `comms_instr` block, see “(Instrumentation Block)” on page 92. For information about the related `offset_comms_instr` block, see “gmsk” on page 98.

Figure 1-47 on page 102 shows how the `offset_comms_instr` and `instr_term` blocks should be used. The `comms_instr` block is used similarly. The circuit consists of two branches driven from a single baseband signal generator. The top branch is the non-ideal receiver model, the bottom branch is an ideal version of the top branch. The ideal version is ideal as you like. The ideal branch computes ideal symbol locations in the complex plane. The instrumentation block compares ideal and non-ideal symbols to compute error-vector-magnitude.

Figure 1-47 EVM setup



© 1994-2020
All Rights Reserved.

offset_comms_instr

(Instrumentation Block)

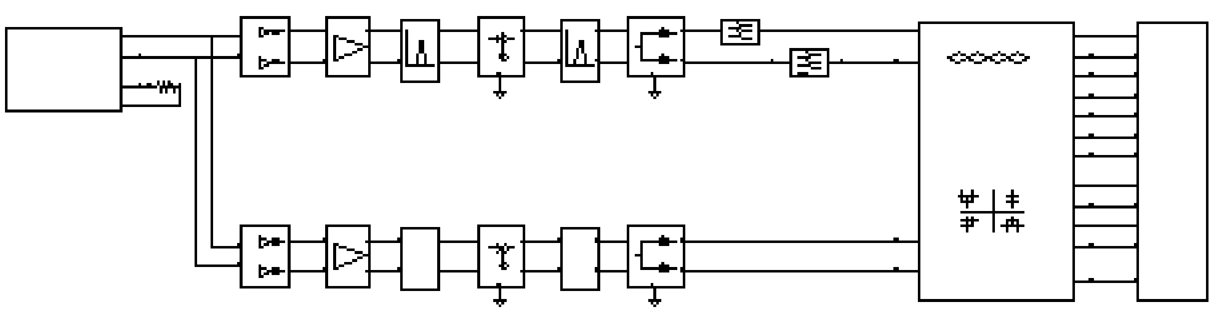
The `offset_comms_instr` block generates waveforms that can be used to create eye-diagrams, eye-diagram statistics, scatter plots, and rms error-vector-magnitudes.

The offset block is identical to the `comms_instr` except that the sampling time for scatter plots and eye-diagram statistics are delayed by half a symbol period. The delay makes it possible to plot symbols in an offset QPSK modulation scheme.

For information about the parameters of the block, see “Instrumentation Block Parameters” on page 93. For information about the outputs of the block, see “Instrumentation Block Outputs” on page 94. For information about the related `comms_instr` block, see “(Instrumentation Block)” on page 92. For information about the related `instr_term` block, see “instr_term” on page 102.

Figure 1-48 on page 103 shows how the `offset_comms_instr` and `instr_term` blocks should be used. The circuit consists of two branches driven from a single baseband signal generator. The top branch is the non-ideal receiver model, the bottom branch is an ideal version of the top branch. The ideal version is as ideal as you like. The ideal branch computes ideal symbol locations in the complex plane. The instrumentation block compares ideal and non-ideal symbols to compute error-vector-magnitude.

Figure 1-48 EVM setup



© 1994-2020 All Rights Reserved.

pi_over4_dqpsk

(Pi/4-DQPSK Signal Source)

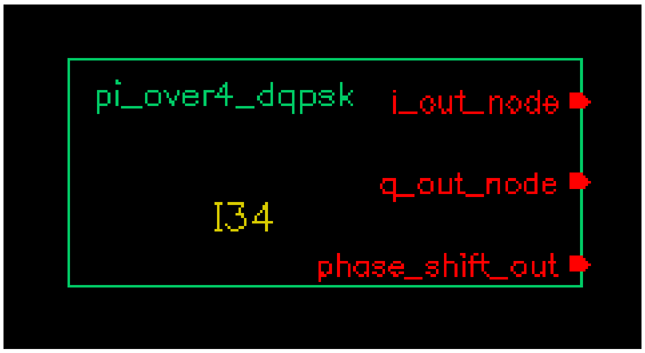


Figure 1-49 shows the block diagram for this source.

Figure 1-49 Π/4-DQPSK baseband signal generator

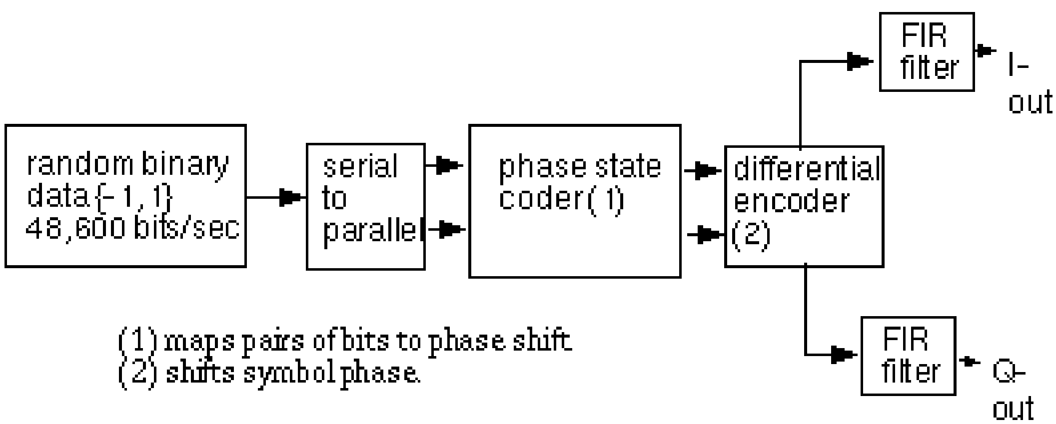


Table 1-50 shows how the phase shift is generated.

Figure 1-50 Phase Shift

1st bit	2nd bit	Phase shift
0	0	$\pi/4$
0	1	$3\pi/4$

© 1994-2020 All Rights Reserved.

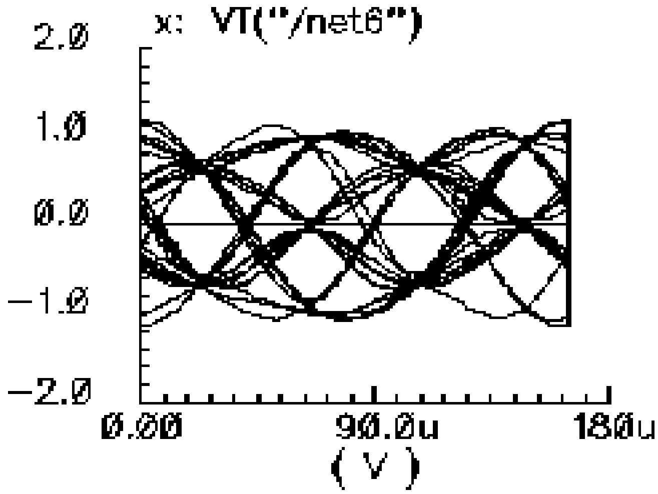
1st bit	2nd bit	Phase shift
1	0	$-\pi/4$
1	1	$-3\pi/4$

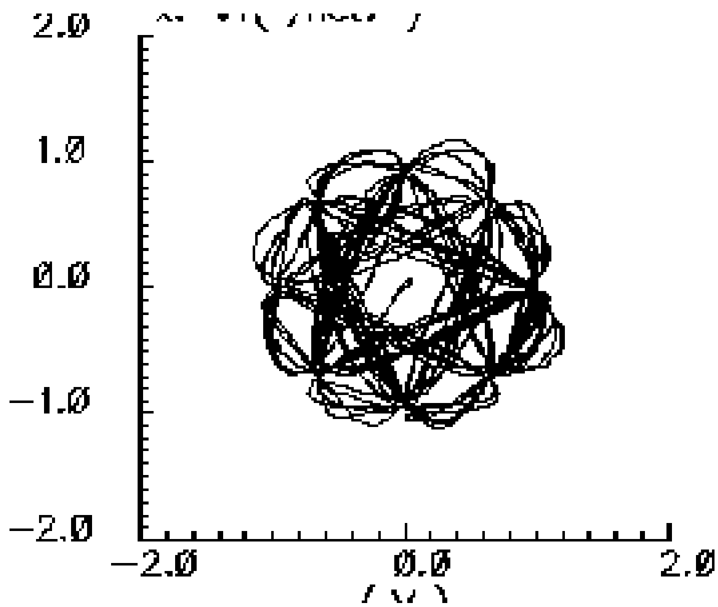
The symbol rate is 24300 symbols per second and the sample rate is 8 times that. The FIR filter is a raised cosine filter implemented with 64-taps.

The eye-diagram generator (`eye_diagram_generator`) created the eye-diagram and trajectory. Figure 1-51 shows the eye-diagram and trajectory for this generator.

© 1994-2020 All Rights Reserved.

Figure 1-51 Eye Diagram and Pi/4 Trajectory





The `amplitude` parameter lets you set the amplitude of the unfiltered signals. An amplitude of “1” means that each FIR filter is driven by 1-volt impulses. If you change the internal variable `IMPULSE_PULSE` to 2, the filters are driven by 1-volt pulses of four samples duration.

The `seed` parameter lets you change the random number generator seed.

© 1994-2020 All Rights Reserved.

Pi/4-DQPSK Signal Source Outputs

The generator creates three output signals.

`i_out_node`	The phase, multiplied by the amplitude.
`q_out_node`	The phase, multiplied by the amplitude.
`phase_shift_out`	The phase shift from one symbol to the next.

Changing the FIR filter in a Pi/4-DQPSK Signal Source

You cannot change the FIR filter, such as the tap length and tap coefficients, directly from the instance. However, you can do so using the Modelwriter as described in "Modifying the BB Signal Generators Using Modelwriter" on page 294.

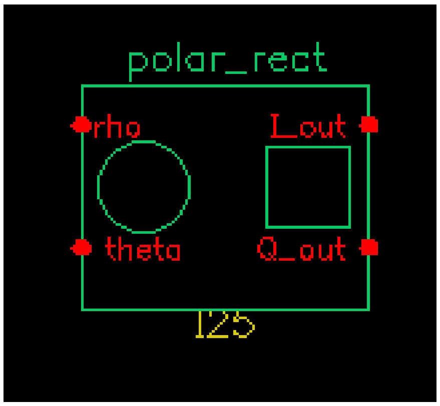
Pi/4-DQPSK Signal Source Output Transitions

The filtered outputs slew linearly from one value to the next because the rise and fall times in the transition statements equal one period. To make the outputs take abrupt steps, copy the module to your library and change the rise and fall times in the last transition statements.

© 1994-2020 All Rights Reserved.

polar_rect

(Polar-to-Rectangular Transformation)



The polar-to-rectangular block is in the measurement category. The only parameters are input and output resistances. The inputs are the baseband signal in polar coordinates, the outputs are the baseband signal in rectangular coordinates.

- Inputs = ρ and θ volts.
- Outputs = i and q volts, such that

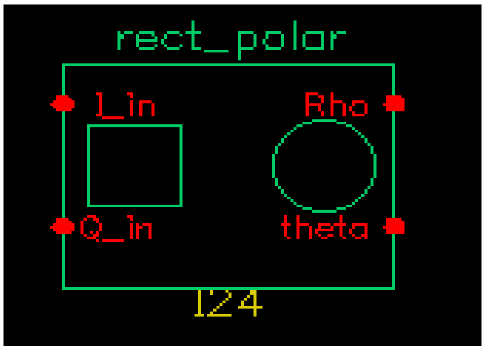
$$i = \rho * \cos(\theta)$$

$$q = \rho * \sin(\theta)$$

© 1994-2020
All Rights Reserved.

rect_polar

(Rectangular-to-Polar Transformation)



The rectangular-to-polar block is in the measurement category. The only parameters are input and output resistances. The inputs are the baseband signal in Cartesian coordinates, the outputs are the baseband signal in polar coordinates.

Parameters: Input and output resistances.

- Inputs = `i` and `q` volts.
- Outputs = ρ and θ volts, such that

$$\rho = \sqrt{i*i + q*q}$$

and

$$\theta = \mathrm{atan}[q/i] + \pi*(1 + \mathrm{sgn}[i])/2$$

where θ is in radians and with appropriate checks for the $i = 0$ case

© 1994-2020
All Rights Reserved.

Passband_components Category

The `Passband_components` category contains the following elements, discussed in the sections that follow.

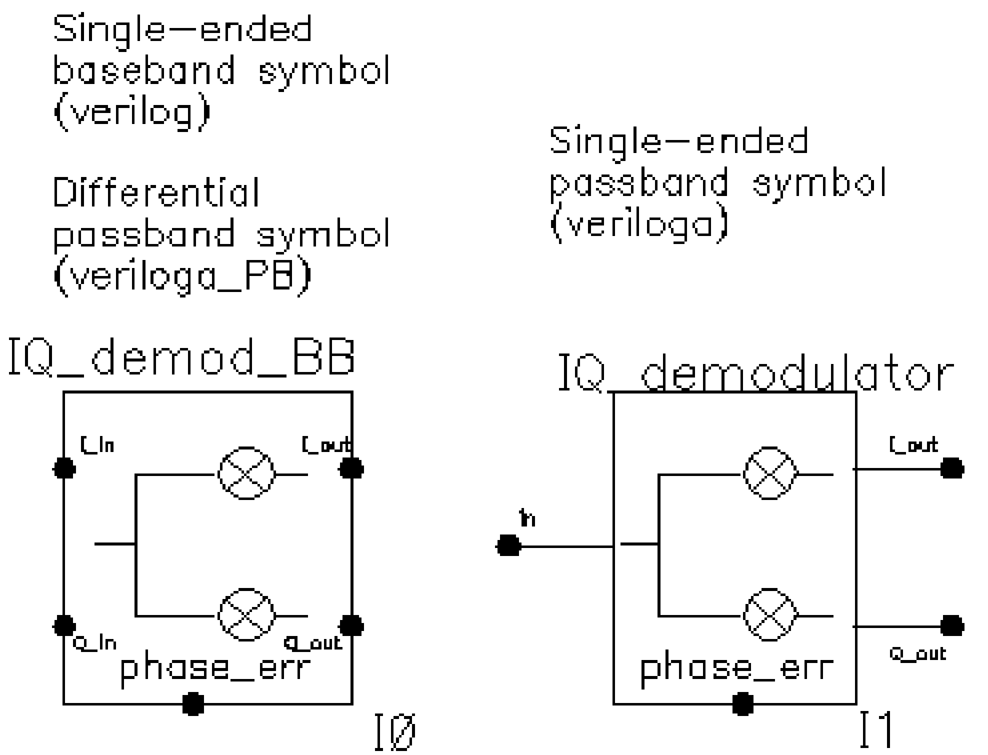
- IQ_demodulator
- IQ_modulator
- LNA_PB
- MIXER_PB
- PA_PB
- shifter_combiner
- shifter_splitter

© 1994-2020 All Rights Reserved.

IQ_demodulator

`(passband = IQ_demodulator)`

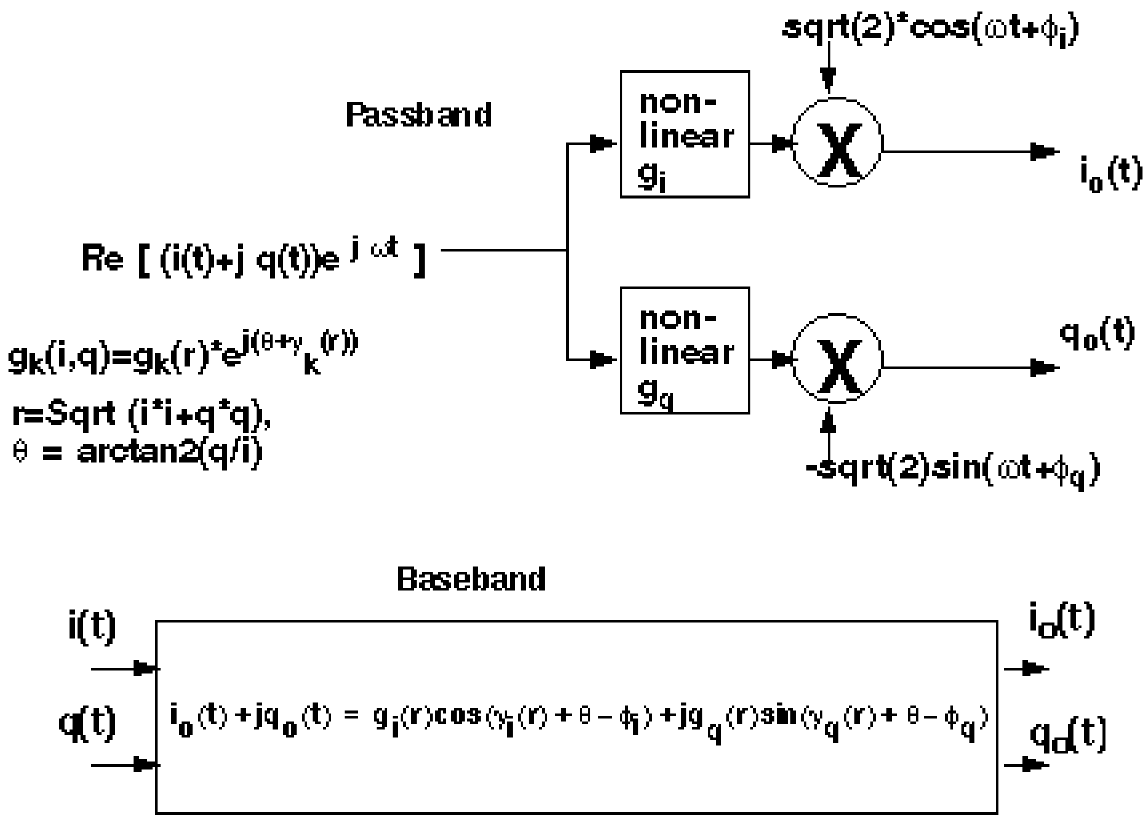
Figure 1-52 Baseband and Passband IQ Demodulator Models



The `IQ_demodulator` converts RF (or IF) to baseband. Figure 1-53 on page 112 shows exactly what the passband demodulator model does. The parameters are like those in the modulator blocks except saturation is specified by input referred IP3 instead of by 1 dB compression point. IP3 was chosen over the 1 dB compression point for specifying saturation because the demodulator usually lies in the receive path and receiver blocks are usually specified with IP3.

© 1994-2020 All Rights Reserved.

Figure 1-53 IQ Demodulator Calculations



The parameters are:

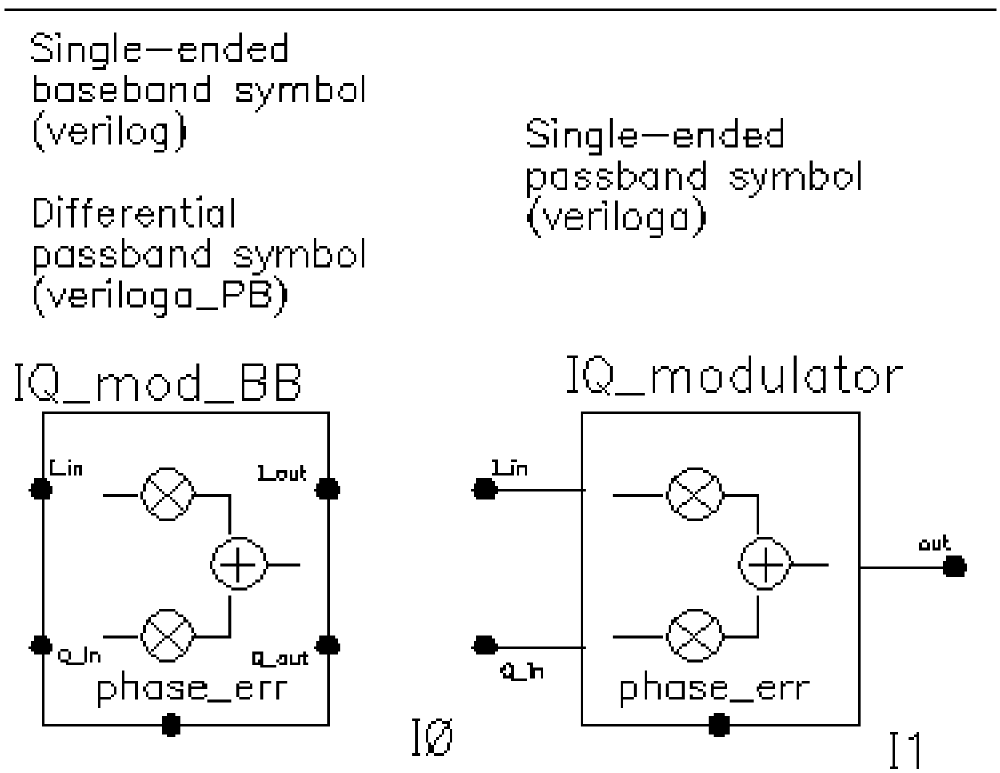
`flow`	Local oscillator frequency.
`I_gain`	Available power gain [dB].
`I_IP3`	Input referenced IP3 (dbm].
`nf`	Noise figure [dB].
`Q_gain`	Voltage gain [dB].
`Q_IP3`	Input referenced IP3 [dBm].
`quad_error`	Quadrature error.
`rin`	Input resistance.
`rout`	Output resistance.

© 1994-2020 All Rights Reserved.

IQ_modulator

```
(passband = IQ_modulator)
```

Figure 1-54 Baseband and Passband IQ Modulator Models



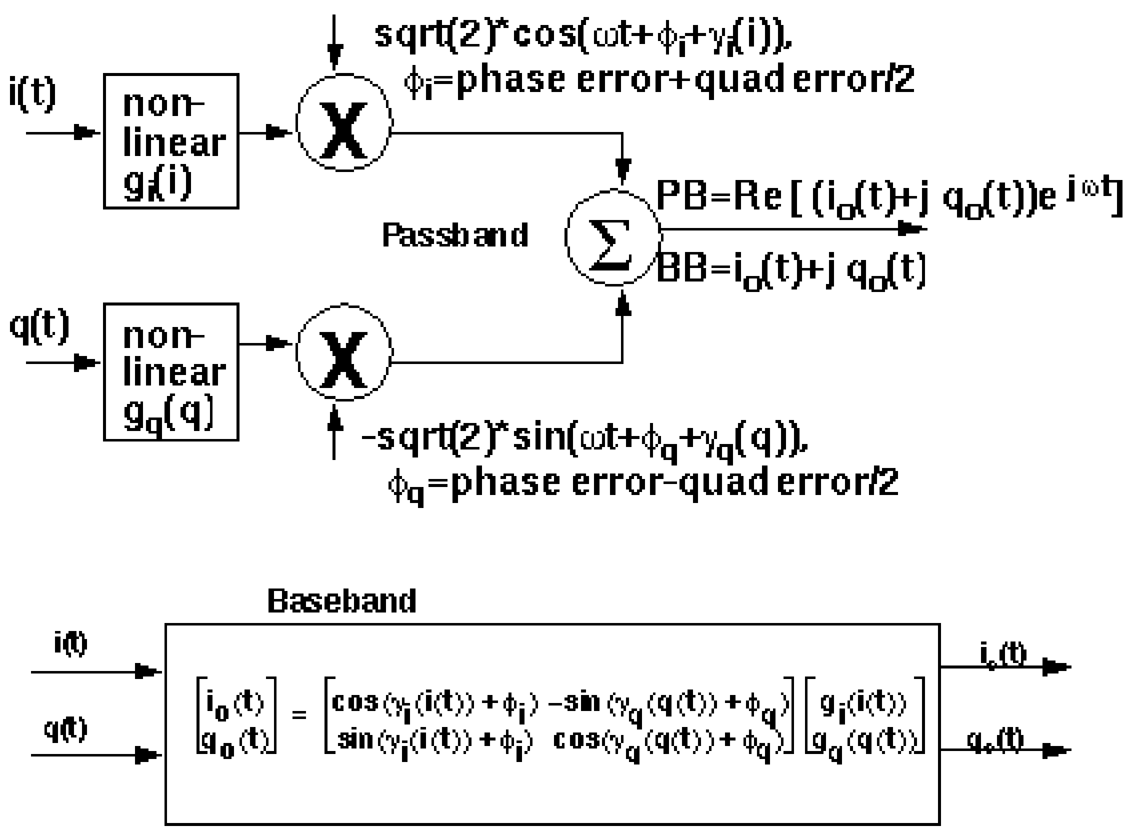
The `IQ_modulator` converts baseband signals to RF or IF. Figure 1-55 on page 114 summarizes exactly what the passband IQ modulator model does. The only difference between the baseband and passband models is carrier suppression. The non-linear functions, g_i and g_q, are specified by their available power gain and 1dB compression points just as in the power amplifier. The functions γ_i and γ_q characterize AM/PM effects in each mixer and are specified by the same parameters that specify power amplifier AM/PM conversion. Because noise is always added at the input, and the input is at baseband in this case, the noise sources are not doubled as they are in the power amplifier or LNA models. Noise figure is defined with reference to one input. Noise is injected at both inputs but the noise injected at just one input alone produces the specified noise figure. Thus, the noise figure parameter should be interpreted as noise figure per input. This model also includes a parameter called `quadrature error` which specifies how far away the two local oscillators signals are from being exactly in quadrature.

© 1994-2020
All Rights Reserved.

`Phase error` is the voltage on the phase error pin. The phase error pin has a fixed noiseless resistive input impedance of 50 ohms. The phase error pin can be used to introduce a dynamic phase error or phase noise. Phase noise can be fed into the phase error pin from a phase-domain PLL model or from a Port. Noise in Port models can be specified either by the internal resistance or by a data file that tabulates a power spectral density. The phase error pin can also be driven by a ramp or circular integrator output to model a frequency error between the incoming carrier and local oscillator.

The following parameters specify the IQ modulator. The available power gain and one dB compression point are explained first. The effects of the phase_error pin and the quadrature error parameter are discussed at the end of this section.

Figure 1-55 IQ Modulator Calculations



$$\begin{bmatrix} i_o(t) \\ q_o(t) \end{bmatrix} = \begin{bmatrix} \cos(\gamma_i(i(t)) + \phi_i) & -\sin(\gamma_q(q(t)) + \phi_q) \\ \sin(\gamma_i(i(t)) + \phi_i) & \cos(\gamma_q(q(t)) + \phi_q) \end{bmatrix} \begin{bmatrix} g_i(i(t)) \\ g_q(q(t)) \end{bmatrix}$$

The parameters are:

	Quadrature error.
`flo`	Local oscillator frequency.

© 1994-2020 All Rights Reserved.

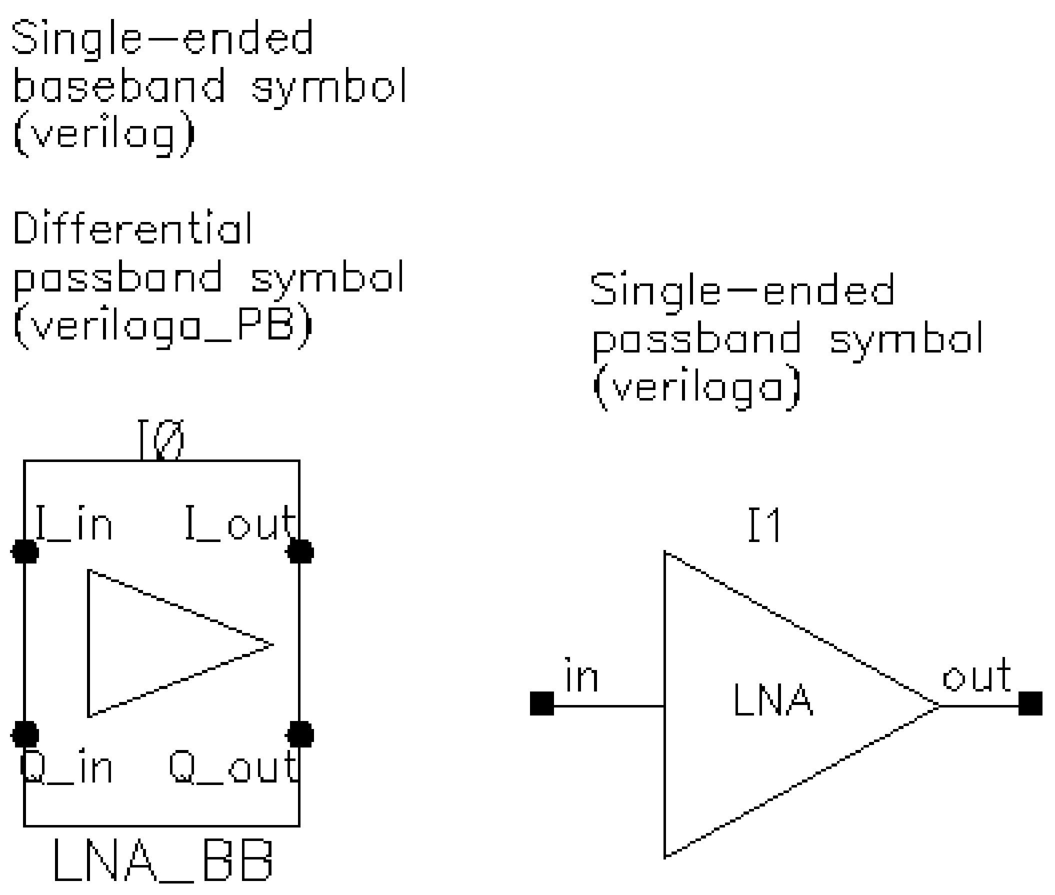
`I_cpdb`	I-output 1dB CP [dBm].
`I_gain`	Available I-mixer gain [dB].
`nf`	Noise figure [dB].
`Q_cpdb`	Q-output 1dB CP [dBm].
`Q_gain`	Available Q-mixer gain [dB].
`rin`	Input resistance.
`rout`	Output resistance.

© 1994-2020 All Rights Reserved.

LNA_PB

`(passband = LNA_PB)`

Figure 1-56 Baseband and Passband Power Amplifier Models



The following parameters specify the low noise amplifier model.

The parameters are:

`cmp`	Input power point for phase point [dBm].
`cw`	Determines the direction of the phase shift. The phase shift is only in one direction. +1 means counter-clockwise, -1 means clockwise, and 0 means no phase shift (no am/pm conversion).
`gain`	Available power gain.

© 1994-2020
All Rights Reserved.

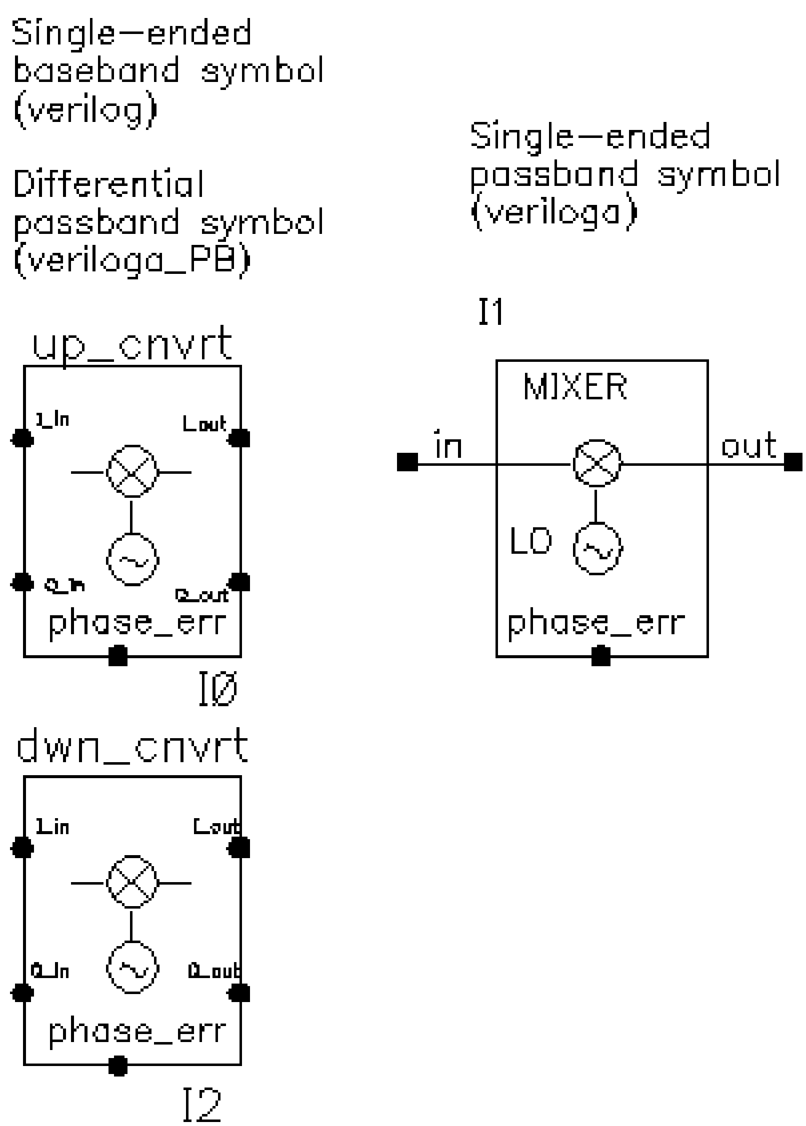
`IP3`	Input referred IP3 [dBm].
`nf`	Noise figure.
`pscp`	Output phase shift at cmp [radians].
`psinf`	Output phase shift as the input power goes to infinity.
`rin`	Input resistance.
`rout`	Output resistance.
`shp`	Determines how fast the phase shift occurs with increasing input power. A larger number delays the shift but makes the shift rise faster as a function of input signal level.

© 1994-2020 All Rights Reserved.

MIXER_PB

`(passband = MIXER_PB)`

Figure 1-57 Baseband and Passband Mixer Models



`MIXER_PB` is a passband model that converts RF to IF and IF to RF.

© 1994-2020 All Rights Reserved.

The parameters are:

`flo`	Local oscillator frequency.
`gain`	Available power gain [dB].
`IP3`	Input referenced IP3 [dBm].
`nf`	Noise figure [dB].
`psinf`	Output phase shift as the input power goes to infinity.
`rin`	Input resistance.
`rout`	Output resistance.

© 1994-2020 All Rights Reserved.

PA_PB

The parameters are:

`cpdb`	Output referenced 1dB compression [dBm].
`gain`	s21 referenced [dB].
`nf`	Noise figure [dB].
`rin`	Input impedance [Ohm].
`rout`	Output impedance [Ohm].

© 1994-2020 All Rights Reserved.

shifter_combiner

The `shifter_combiner` combines two signals so that they add if one leads the other by 90 degrees and so that they cancel if it lags by 90 degrees.

The parameters are:

`freq`	Frequency at which the magnitudes are balanced.
`gain`	Multiplies the input voltages.
`r`	Impedance of the internal resistor.
`rin`	Input terminal impedances.
`rout`	Output impedance.

© 1994-2020 All Rights Reserved.

shifter_splitter

The `shifter_splitter` splits a signal into two signals 90 degrees out of phase with each other.

The parameters are:

`freq`	Frequency at which the magnitudes are balanced.
`gain`	Multiplies the input voltages.
`r`	Impedance of the internal resistor.
`rin`	Input impedance.
`rout`	Output impedance.

© 1994-2020 All Rights Reserved.

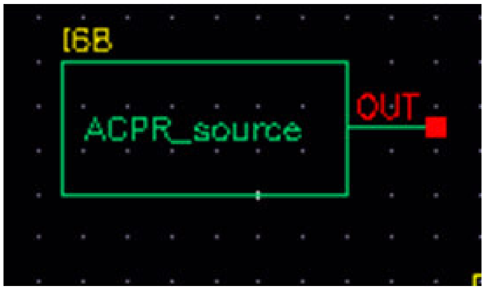
RF_components Category

The cells in the `RF_components` category are:

- ACPR_source
- balun
- balun_com
- gfsk
- lna
- mixer
- ofdm
- osc
- pa
- quadrature
- RFVsource
- shifter
- Triplexer
- Ten_plexer

© 1994-2020 All Rights Reserved.

ACPR_source



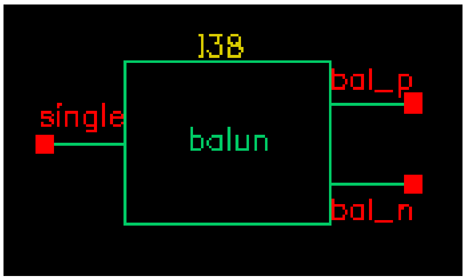
The `ACPR_source` is used to generate modulated RF waveforms. It is a self-contained RF modulator. The out connection is the modulated RF output.

The parameters are:

`I PWL file name`	File name for the I modulation piecewise linear file
`Q PWL file name`	File name for the Q modulation piecewise linear file
`Gain`	Linear multiplier for the amplitude
`RF Frequency`	RF carrier frequency
`RF Amplitude (dBm)`	RF amplitude in dBm
`Reference Resistance`	Output resistance (Ohms)

© 1994-2020 All Rights Reserved.

balun



The balun (balancing transformer) is used in circuits that require single/differential signal transformation. Although a passive network (including the transformer) is used to achieve balun, this implementation employs a three-port network. There are three ports (or nodes), because the reference nodes are always at the global ground: `single`, `bal_p`, and `bal_n`.

The three ports are

`single`	Single end.
`bal_p`	In-phase end of the balanced output.
`bal_n`	Out-of-phase end of the balanced output.

When the ports are numbered as `single(1)`, `bal_p(2)`, and `bal_n(3)`, the S-parameter for the three-port network is

$$S = \begin{bmatrix} 0 & t & -t \\ t & 0 & 0 \\ -t & 0 & 0 \end{bmatrix}$$

where

$$t = \frac{10^{-loss/10}}{\sqrt{2}}$$

when loss is specified in dB.

This module can also be used in common mode cancellation applications.

© 1994-2020 All Rights Reserved.

The module is declared as follows

```
module balun(single, bal_p, bal_n);
  inout single, bal_p, bal_n;
  electrical single, bal_p, bal_n;
   parameter real rin = 50 from (0:inf);
   parameter real rout = 50 from (0:inf);
   parameter real loss = 0 from [0:inf);
```

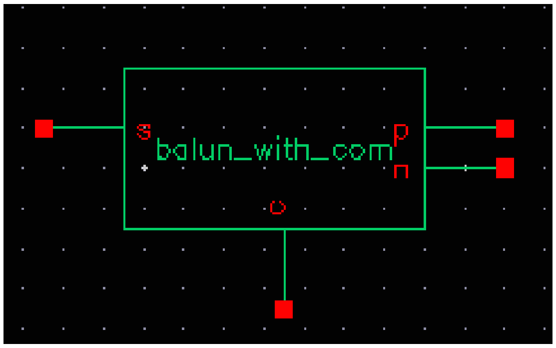
Parameters include the input impedance (for single end), the output impedance (for balanced end to ground), and the insertion loss (from single end to balanced end and from balanced end to single end).

The parameters are:

`loss`	Insertion loss [dB].
`rin`	Input impedance [Ω].
`rout`	Output impedance [Ω].

© 1994-2020 All Rights Reserved.

balun_com



The `balun_com` has, in addition to the three ports of the `balun`, an external reference node that can be used for DC bias set up in the balanced end. The balun_com is equivalent to the balun when the voltage of the reference node `c` is set to 0.

The four ports of the `balun_com` are:

`c`	Common (reference) node for `p` and `n`.
`n`	Out-of-phase end of the balanced output.
`p`	In-phase end of the balanced output.
`s`	Single end.

The module is declared as follows

```
module balun_com(s, p, n, c);
  inout s, p, n, c;
  electrical s, p, n, c;
   parameter real rin = 50 from (0:inf);
   parameter real rout = 50 from (0:inf);
   parameter real loss = 0 from [0:inf);
```

Parameters include the input impedance (for single end), the output impedance (for balanced end to ground), and the insertion loss (from single end to balanced end and from balanced end to single end).

The parameters are:

`loss`	Insertion loss [dB].
`rin`	Input impedance [Ω].
`rout`	Output impedance [Ω].

© 1994-2020 All Rights Reserved.

gfsk

(Gaussian-filtered frequency shift keying I and Q modulation generator)

GFSK is a Gaussian filtered frequency modulation I and Q signal generator, which makes the I and Q output smoother so as to limit the spectral width. GFSK is poorer in terms of the spectral efficiency but easier to implement as compared to GMSK. GFSK is often seen in cordless phones using the DECT standard and in bluetooth.

The symbol connection ports are:

`inbit`	Inbit is a connection for an external random bit stream. This connection is enabled when the `enable_input` property is set to 1.
`iout`	Output I modulation signal.
`qout`	Output Q modulation signal.

Instance Parameters

The parameters are:

Name	Description
`enable_input`	When `enable_input` is set to 1, the digital signal on the inbit terminal is read in as the binary sequence to be modulated. Signals above 0.5 volts are interpreted as the 1 state and signals below 0.5 volts are interpreted as zero state. When `enable_input` is set to 0, an internal random number generator is used for the binary input sequence. Only 0 and 1 are allowed, and the default is 0.

© 1994-2020 All Rights Reserved.

`BT`	Inside the block is a gaussian filter with the bandwidth set based on the symbol rate and the filter bandwidth times the period of the input bit product. The default is 0.5, which is the value for the bluetooth basic rate standard.
`Initial_phase`	Inside the gfsk block, the phase in Radians of the output is integrated. This phase determines the final I and Q output signal. The default is to start at 0 radians. Using the default value is highly recommended.
`Ec`	This sets the power per bit in dB into a 1 ohm load. The default value is -116dBW.
`frame time`	This block was originally created to model the bluetooth basic rate standard. This property sets the total time of each packet. Note that the gfsk component does not add any header or trailer information. If this is desired, it must be incorporated into the external bit stream applied at the input of the gfsk component, and the `enable_input` property should be set to 1. The default is 625usec, which is one slot time of the bluetooth basic rate standard.
`pulse_length`	The number of symbols preloaded into the filter, before it starts to process the data on 'inbit'. This should be set to the default value of 1 and must be an integer. Setting larger values introduces delays and sample averaging that affect the frequency content of the output I and Q signal.
`Samples`	Sets the number of samples in each bit time, and should be left at the default of 100. At each sample point, the output phase is numerically integrated.
`Initial_syms`	The value of the symbols preloaded into the filter. Remove the curly brackets, and use the default value of 1.
`seed`	This is the seed value for the internal random number generator. The default value is 21, and can be set to any integer.
`frame_size`	This defines the number of bits that are contained in one packet. This must be an integer, and is set to 625, which is the total number of bits in the bluetooth basic rate packet.
`index`	This is the modulation index for the FM signal that is generated.
`need_deactivate`	The default is 1, which causes normal operation of the gfsk component. Setting this value to 0 causes the I and Q output to remain constant.
`enable_start`	This property should be left at the default value of 4.

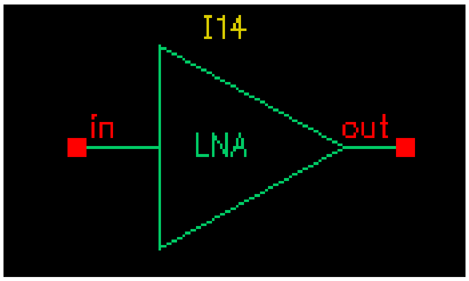
© 1994-2020
All Rights Reserved.

`enable_stop`	This property should be left at the default value of 5.

© 1994-2020 All Rights Reserved.

lna

(Low-Noise Amplifier)



Low-noise amplifiers (LNAs) are commonly used in receiver designs to amplify the signal with a low noise figure. A typical LNA has the following three sets of parameters:

- Linear model
- Nonlinear model
- Noise model.

The module is declared as follows:

```
module lna(in, out);
   inout in, out;
   electrical in, out;
   parameter real nf = 2 from [0:inf);
   parameter real ip3 = -10;
   parameter real gain = 15 from [0:inf);
   parameter real isolation = 200 from (0:inf);
   parameter real rin = 50 from (0:inf);
   parameter real cin = 0 from [0:100];
   parameter real rout = 50 from (0:inf);
   parameter real cout = 0 from [0:100];
   parameter real gammain = -150 from (-inf:0];
   parameter real mismatch = 1 from [-1:1] exclude (-1:1);
   parameter real gammaout=-150 from (-inf:0];
```


© 1994-2020
All Rights Reserved.

The parameters are:

`cin`	Parasitic input capacitance [pF].
`cout`	Parasitic parallel output capacitance [pF].

Note: Although the label in the properties list reads *picofarads*, the actual value should be set in *nanofarads* for `cin` and `cout`.

`gain`	S21 (power gain) [dB].
`gammain`	Input return loss [dB].
`gammaout`	Output return loss [dB].
`ip3`	Input referenced IP3 [dBm].
`isolation`	S12 [dB].
`mismatch`	Mismatch sign of input. 1: input impedance > reference impedance -1: otherwise.
`nf`	Noise figure [dB].
`rin`	Reference impedance of the input port [Ω].
`rout`	Reference impedance of the output port [Ω].

Internally, a set of linear equations is constructed to satisfy the S-parameters. Furthermore, nonlinearity, expressed by a third-order polynomial function, is added to the gain (or S21) to describe the IP3. Excess white noise is added at the input port to describe the noise figure.

IP3 is the measure of the corruption of signals due to the third-order intermodulation of two nearby tones as shown in Figure 1-58. You measure this parameter using a two-tone test. Avoid the measurement of IP3 by a single tone test.

© 1994-2020 All Rights Reserved.

Figure 1-58 Intermodulation of Two Nearby Signals

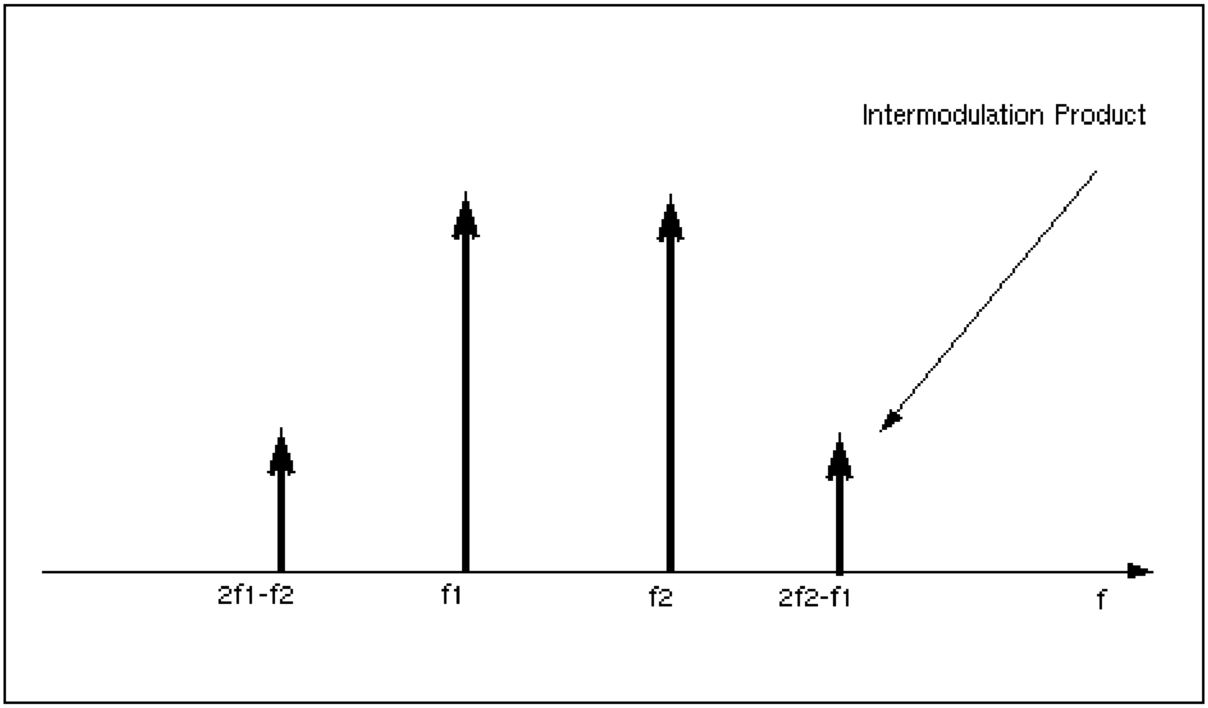
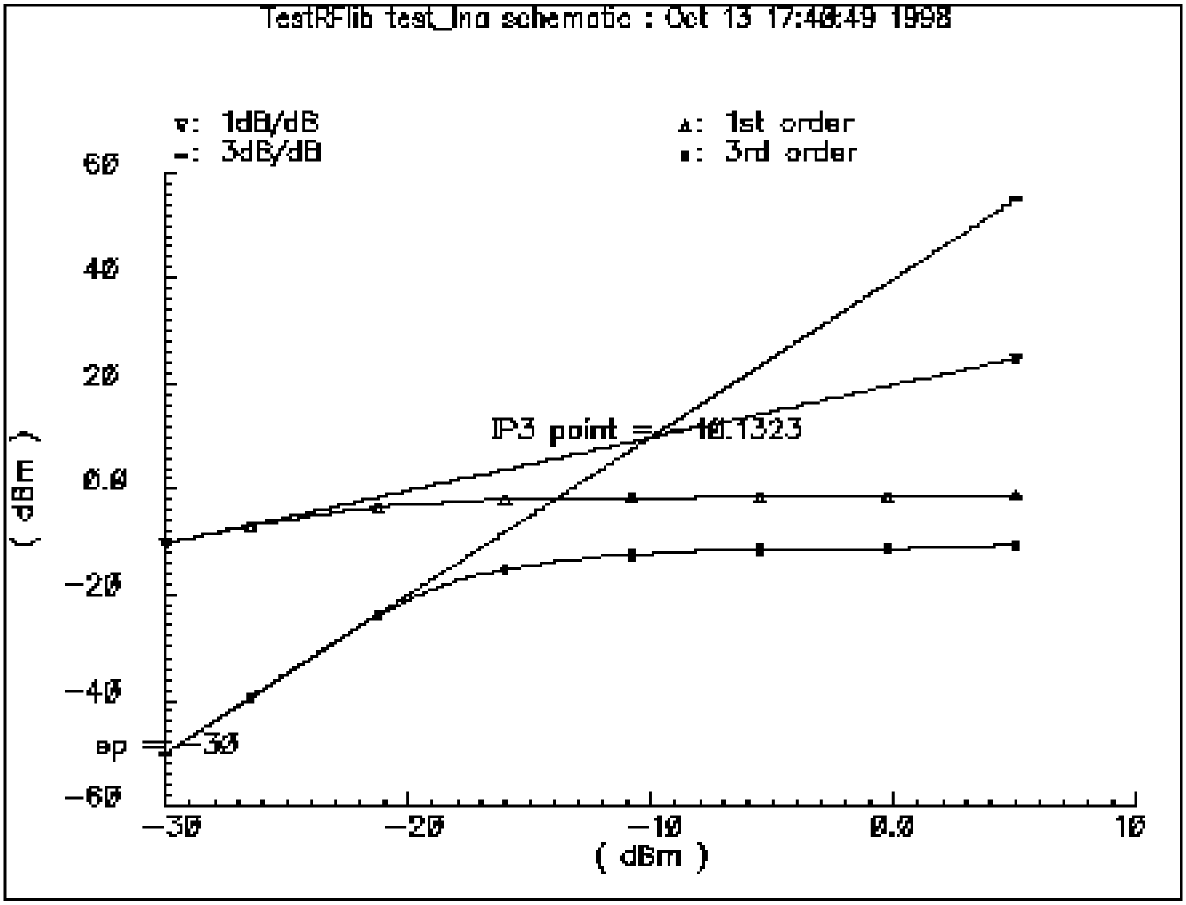


Figure 1-59 shows the captured IP3 when the requested value of IP3 is –10dBm.

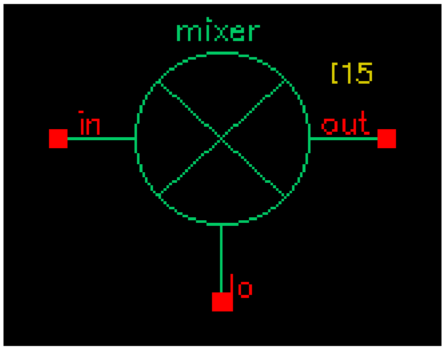
© 1994-2020
All Rights Reserved.

Figure 1-59 IP3 from Spectre RF Simulation



All Rights Reserved.

mixer



Mixers are important for frequency translation in RF circuits. A typical mixer has the following three sets of parameters.

- Time-varying linear model
- Nonlinear model
- Noise model

This RF library model describes the typical behavior of integrated mixers. The LO switches the input signal on and off. Input LO power beyond the specified limit is effectively clipped off.

Declare the module as follows

```
module mixer(in, lo, out);
    electrical in, lo, out;
    parameter real gain = 10 from [-50:50];
    parameter real plo = 10 from [-100:100];
    parameter real rin = 50 from (0:inf);
    parameter real rout = 200 from (0:inf);
    parameter real rlo = 50 from (0:inf);
    parameter real ip2 = 5;
    parameter real ip3 = 5;
    parameter real nf = 2 from [0:inf);
    parameter real isolation_LO2IN = 20 from (0:inf);
    parameter real isolation_LO2OUT = 20 from (0:inf);
    parameter real isolation_IN2OUT = 20 from (0:inf);
```

The parameters are:

`gain`	Gain from IN to OUT [dB].
`ip2`	Input referenced IP2 [dBm].
`ip3`	Input referenced IP3 [dBm].
`isolation_IN2OUT`	Isolation from IN to OUT [dB].

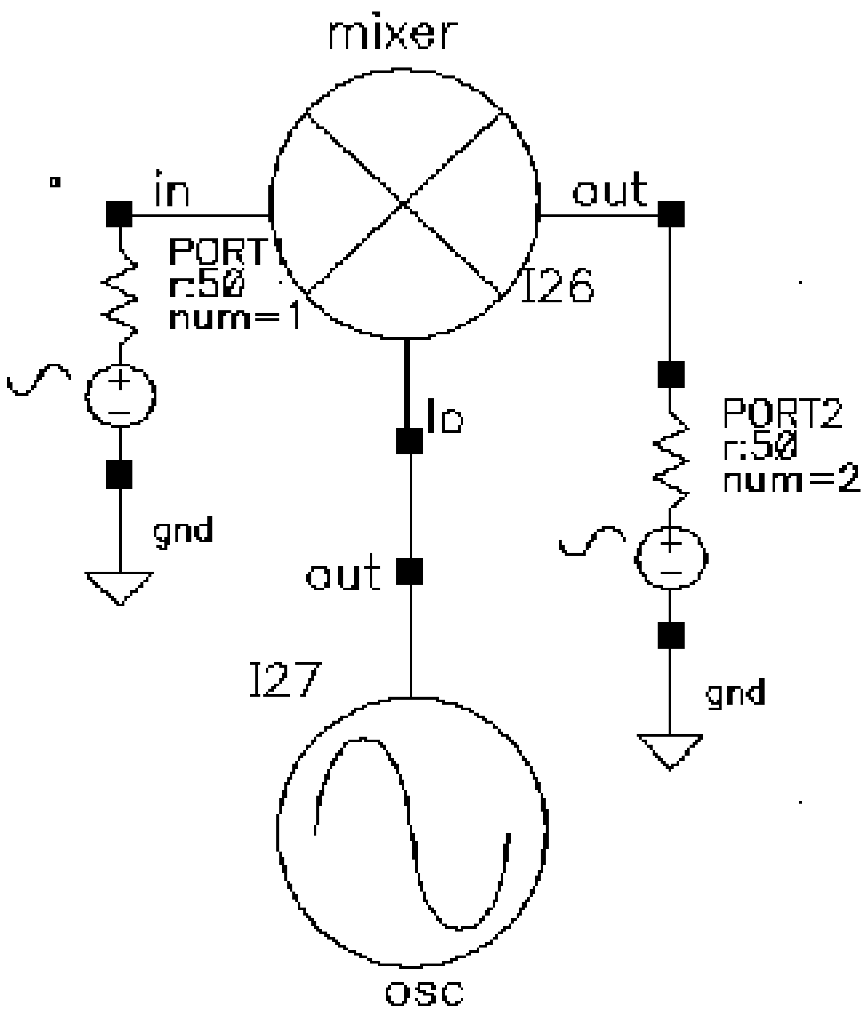
© 1994-2020
All Rights Reserved.

`isolation_LO2IN`	Isolation from LO to IN [dB].
`isolation_LO2OUT`	Isolation from LO to OUT [dB].
`nf`	Noise figure (DSB) [dB].
`plo`	Power of the LO input [dBm].
`rin`	Input impedance for IN [Ω].
`rlo`	Input impedance for LO [Ω].
`rout`	Output impedance for OUT [Ω].

Figure 1-60 is the simple schematic that tests the mixer.

© 1994-2020 All Rights Reserved.

Figure 1-60 Schematic for Testing the Mixer Model



The maximum power of the fundamental frequency of the local oscillator, `plo`, can be used in the mixing process. Therefore, the gain, defined as the output power of the mixed product versus the input power of the RF signal, depends on the power level of the LO. The gain levels off, however, to the specified maximum value as the LO signal becomes larger.

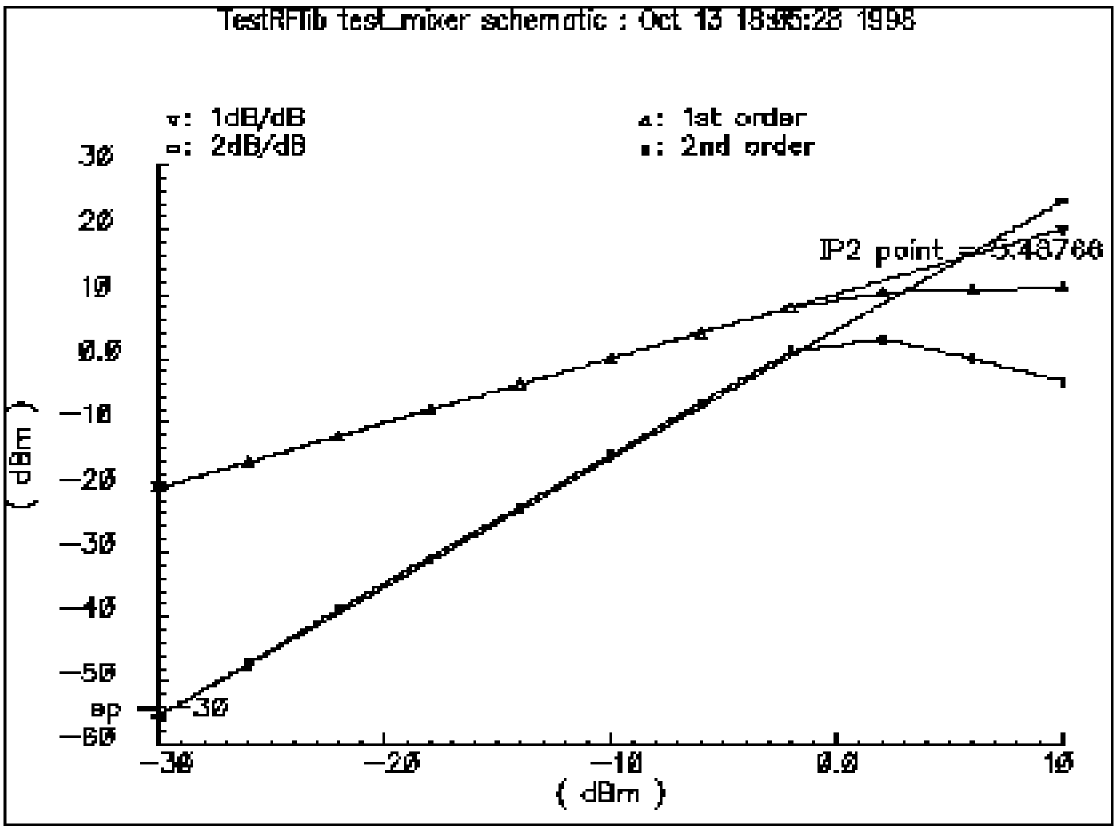
You can measure both IP3 and IP2 with Spectre RF. You must select frequencies carefully when you measure IP3 to measure harmonic distortion (HD) and IP2. Testing IP3 requires two tones to measure the intermodulation distortion (IMD), while testing IP2 requires only one tone.

Assume the RF input frequencies are f_1 and f_2, and the LO frequency is f_{lo}. If the input power level at f_1 equals that at f_2, the IP3 is the intercept point of the extrapolated line of output

© 1994-2020 All Rights Reserved.

power at frequency $|f_{lo} - (2f_2 - f_1)|$ versus the extrapolated line of the linear output signal at $|f_{lo}-f_1|$. Input-referenced IP3, therefore, can be read as the X-axis value at the intercept point. The IP2, for the purpose of measuring the half-IF effects, is defined as the intercept point of the extrapolated line of output power at frequency $|2(f_{lo} - f_1)|$ versus the linear output signal. Figure 1-60 shows that the intercept point of the 1 dB/dB and 2 dB/dB lines is at the X-axis reading of 4.78 dBm, while the requested IP2 value is 5 dBm. The order of the intercept point is based only on the order of the RF signals. The order of LO signal is not counted in the definition of the intercept point. In the implementation of this model, the orders of LO for IP3 and IP2 are 1 and 2 respectively.

IP2 Measurement



Internally, a set of equations is built to satisfy a three-port S-parameter. A third-order polynomial describes the nonlinearity of IP3. The LO signal is further multiplied by itself to derive the second-order harmonic, which is then used to produce the IP2 effect. Excessive white noise is added in the RF input port to satisfy the noise figure. Remember, however, that the noise figure is double-sideband. If the noise at the image frequency is not filtered out, the measured noise figure is 3dB larger than the DSB noise figure.

© 1994-2020
All Rights Reserved.

ofdm

(Orthogonal Frequency-Division Multiplexing)

OFDM is a digital multi-carrier modulation scheme, in which closed-spaced sub-carriers are summed into main carrier. These sub-carriers are orthogonal to each other and modulated with conventional modulation scheme at a low symbol rate. The summation is performed through Fast Fourier Transform.

OFDM has the merit of robost against intersymbol interference and narrow-band co-channel interference. It is spectral-efficient. It has seen many applications such as WiMAX, MBWA, Wi-Fi and UWB and so on.

The ports are

I_in, Q_in [v]	Input signals, activated only when “input_enable” is set to 1.
I_out, Q_out [v]	Output signals, i.e., ofdm baseband signals.

Instance Parameters

The parameters are:

`frame_time`	Frame time
`samples`	Samples in one frame
`Poly_length`	The length of vector for shift register's feedback connections
`Shift_length`	The length of vector for the delay of PN sequence
`Poly_order`	The order of polynomial function
`Init_state_size`	The length of vector of initial state

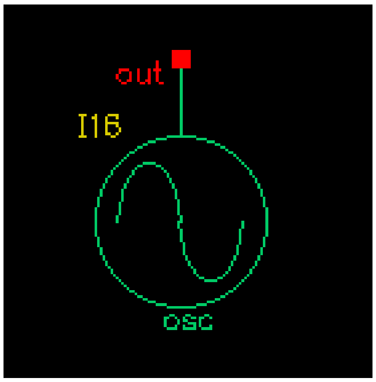
© 1994-2020 All Rights Reserved.

`poly`	The array of polynomial function
`state`	The array of initial state
`shift`	The array for delay of PN
`Dump_frames`	How many frames are skipped initially

© 1994-2020 All Rights Reserved.

osc

(Oscillator)



Oscillator models describe the essential information for a typical oscillator, more precisely, a local RF power source.

The definition of the model in the Verilog-A language is as follows:

```
module osc(out);
    electrical out;
    inout out;
    parameter real power = 10;
    parameter real f = 1e9 from (0:inf);
    parameter real rout = 50 from (0:inf);
    parameter real floor = -60 from (-inf:0);
    parameter real f1 = 1000 from (0:1e6);
    parameter real n1 = -40 from (bottom:0);
    parameter real fc = 0 from [0:f1);
```

The parameters are:

`bottom`	Noise floor [dBc/Hz].
`f1`	Frequency point for `n1` [Hz].
`fc`	Corner frequency of white phase and flicker phase [Hz].
`freq`	Output frequency [Hz].
`n1`	Phase noise at `f1` [dBc/Hz].
`power`	Output power when matched [dBm].
`rout`	Output impedance [W].

© 1994-2020 All Rights Reserved.

This model is not an autonomous model. Rather, it simply generates a sinusoidal wave with the specified impedance, power level, and phase noise characteristics.

When the load is matched to the internal impedance, the load dissipates the specified output power. You can specify the noise floor of the output signal. Furthermore, by adding one point (frequency, phase noise), you can specify $1/f^2$ frequency noise (corresponding to the phase noise induced by white noise). If f_c, the corner frequency of white phase and flicker phase noise, is bigger than 0, $1/f^3$ frequency noise (flicker-noise-induced phase noise) is further specified. Otherwise, $1/f^3$ noise is not included.

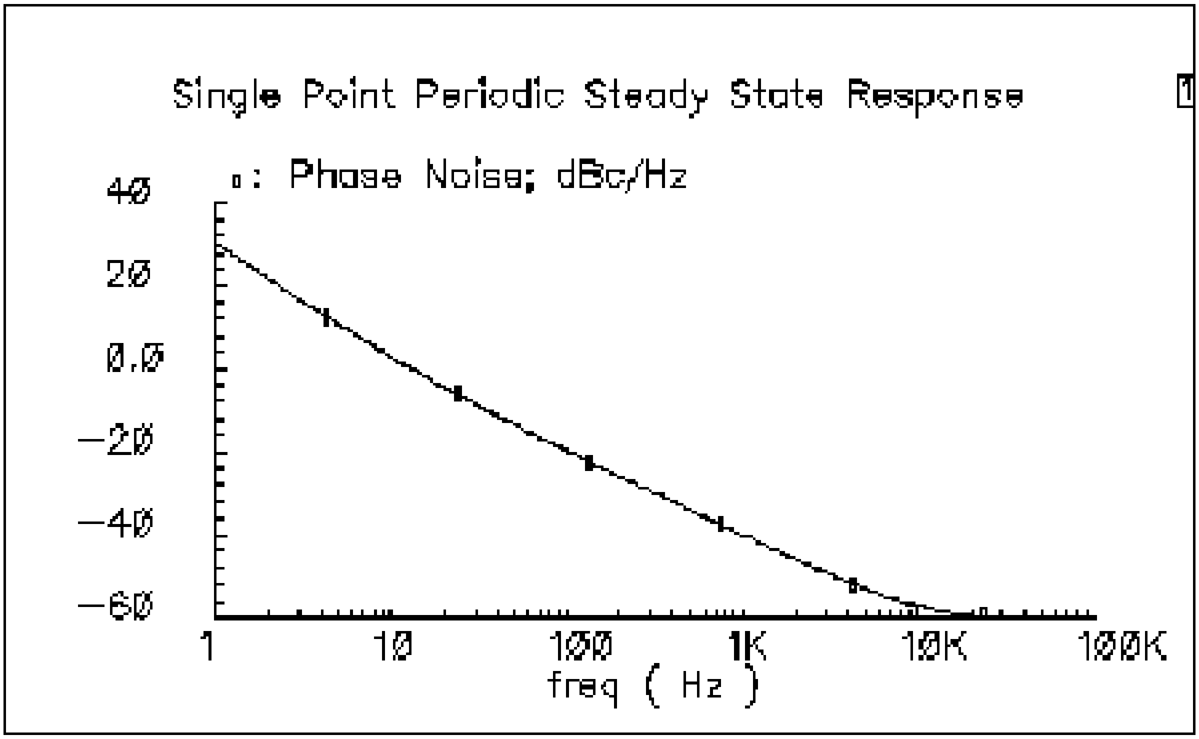
The phase noise values that are symmetric around the carrier are correlated. The noise floor, however, is not correlated.

Figure 1-61 shows the phase noise of the oscillator model. In Figure 1-61, the specified parameters are:

`noise floor`	–60 dBc/Hz
f_1	1 K
n_1	–40 dBc/Hz
f_c	100

© 1994-2020 All Rights Reserved.

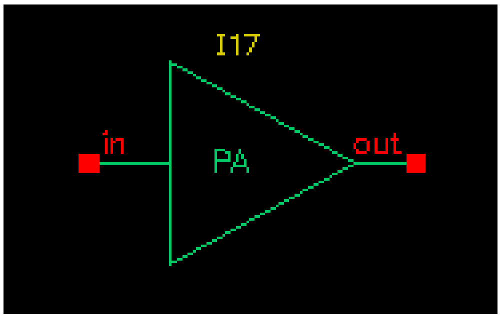
Figure 1-61 Phase Noise for the Oscillator



© 1994-2020
All Rights Reserved.

pa

(Power Amplifier)



Power amplifiers (PAs) are used in RF transmitters to achieve output of a higher power level. The PA model differs from the LNA model in that it has greater power delivery capabilities with less stress on matching capabilities.

The Verilog-A module is declared as follows:

```
module pa(in, out);
    inout in, out;
    electrical in, out;
    parameter real nf = 2 from [0:inf);
    parameter real gain = 20 from [0:inf);
    parameter real rin = 50 from (0:inf);
    parameter real rout = 50 from (0:inf);
    parameter real p1db = 30;
    parameter real psat = 35;
    parameter real ip2 = 40;
```

The parameters are:

`gain`	S21 [dB].
`ip2`	Input-referenced IP2 [dBm].
`nf`	Noise figure [dB].
`p1db`	Output-referenced 1dB compression [dBm].
`psat`	Maximum output power [dBm].
`rin`	Input impedance [Ω].
`rout`	Output impedance [Ω].

The power amplifier model has the following three parts:

© 1994-2020 All Rights Reserved.

- the linear model
- the nonlinear model
- the noise model

Internally, for simplicity, the reverse isolation is assumed to be ideal. A set of linear equations is constructed to satisfy these S-parameters. Nonlinear effects are added to the gain to describe the nonlinearity. The output power of the power amplifier compresses to 1 dB less than the output of an ideal linear amplifier at the 1 dB compression point. Further increase of the input power makes the output approach the saturation power only at the fundamental operating frequency. IP2 describes the second order effects of the amplifier, so use only one tone in the test. Excess white noise is added at the input port to describe the noise figure.

The implementation of `psat` assumes a pure sinusoidal waveform. To maintain a restrained output power, the output waveform is clipped from a sinusoidal to a square wave form. Figure 1-62 shows the input and output waveforms of the power amplifier. Because of the output waveform clipping, the input sinusoidal wave should have a DC component of zero.

Figure 1-62 Input and Output Waveforms of the Power Amplifier

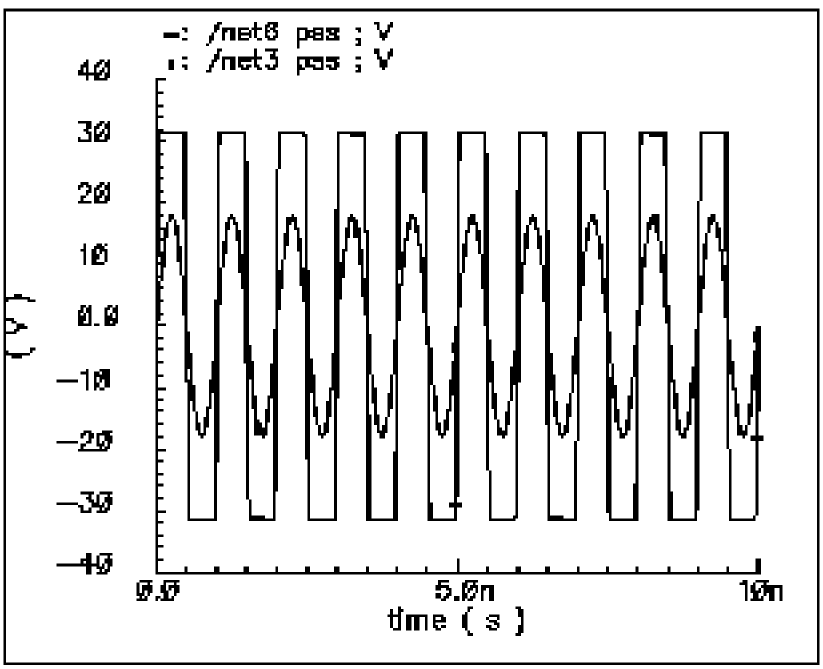
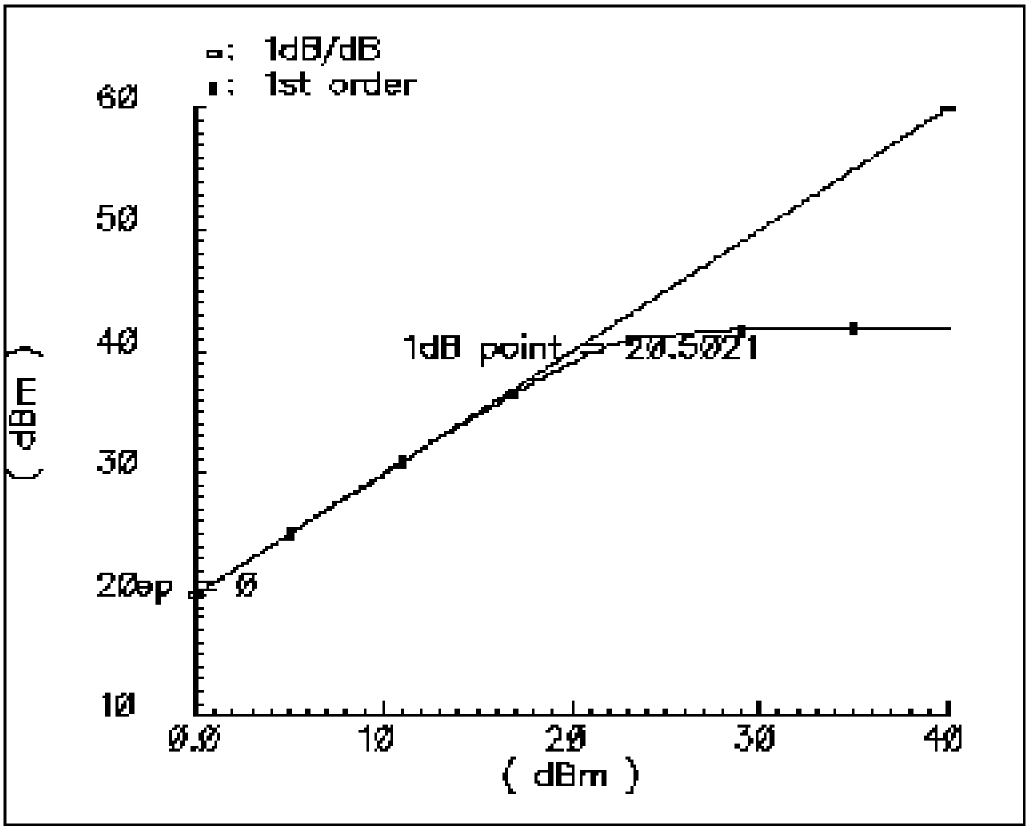


Figure 1-63 shows the 1 dB compression point and the saturation power. This difference is caused by the 50 Ω load impedance. The specified output referenced 1 dB compression point is 40 dBm, which Spectre RF captures as 39.6.

© 1994-2020 All Rights Reserved.

If `psat` is much larger than `p1db`, your `psat` might not be satisfied.

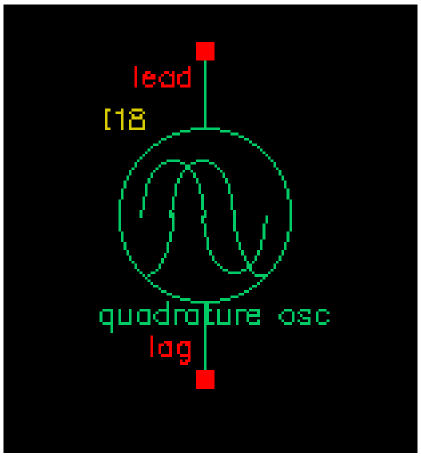
Figure 1-63 1dB Compression Point and Saturation Power



© 1994-2020
All Rights Reserved.

quadrature

(Quadrature Signal Generator)



The quadrature signal generator model is included because, in quadrature receiver design, a phase shifter is ordinarily used to generate the quadrature signal from one signal source such as the VCO. However, a phase shifter is hard to implement in a wide band model.

A quadrature signal consists of two signals with a 90-degree phase difference but with identical noise and amplitude.

The Verilog-A module is declared as follows.

```
module quadrature(lead, lag);
    electrical lead, lag;
    inout out_cos, out_sin;
    parameter real power = 10;
    parameter real f = 1e9 from (0:inf);
    parameter real rout = 50 from (0:inf);
    parameter real floor = -60 from (-inf:0);
    parameter real f1 = 1000 from (0:1e6);
    parameter real n1 = -40 from (bottom:0);
    parameter real fc = 0 from [0:f1);
```

The parameters are:

`bottom`	Noise floor [dBc/Hz].
`f1`	Frequency point for `n1` [Hz].
`fc`	Corner frequency of white phase and flicker phase [Hz].
`freq`	Output frequency [Hz].
`n1`	Phase noise at `f1` [dBc/Hz].

© 1994-2020 All Rights Reserved.

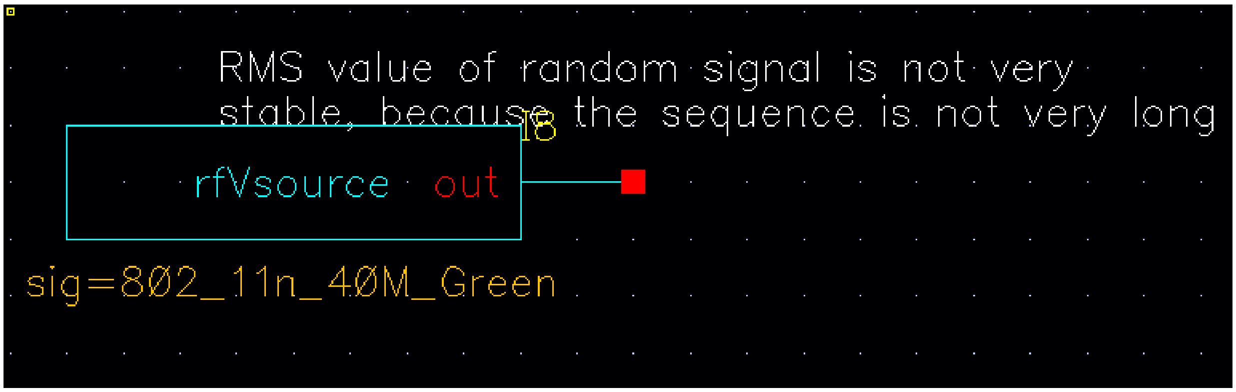
`phase_shift`	
`power`	Output power when matched [dBm].
`rout`	Output impedance [W].

The difference between the quadrature signal generator model and the oscillator model is that the oscillator has only one output node but the quadrature signal generator has two output nodes, `lead` and `lag`. In the quadrature signal generator model, when the power levels, output impedances, and noise sources are identical, the two outputs, `lead` and `lag`, have a 90-degree phase difference.

© 1994-2020 All Rights Reserved.

RFVsource

(RF modulated source for 802.11 standards)



`Sig_Standard`	The signal standard to be used. Choices are: ■ 802.11a, 802.11G-ERP, 802.11n-20M-Mix ■ 802.11n-20M-Legacy, 802.11n-20M-Green, ■ 802.11n-40M-Mix, 802.11n-40m-Legacy, ■ And 802.11n-40M-Green
`Carrier Frequency (Hz)`	RF Carrier frequency in Hz
`Power (dBm)`	RF Power in dBm
`Resistance (Ohms)`	System resistance in ohms.
`Filter`	Choices are none or erc (Raised Cosine)
`Roll off factor`	If erc is chosen for the filter, this parameter is displayed, and this is the rolloff factor of the raised cosine filter. The default is 0.23
`Band width (Hz)`	If erc is chosen for the filter, this parameter is displayed and this sets the bandwidth of the filter. If ACPR is to be measured, set this to twice the bandwidth of the main channel.

© 1994-2020
All Rights Reserved.

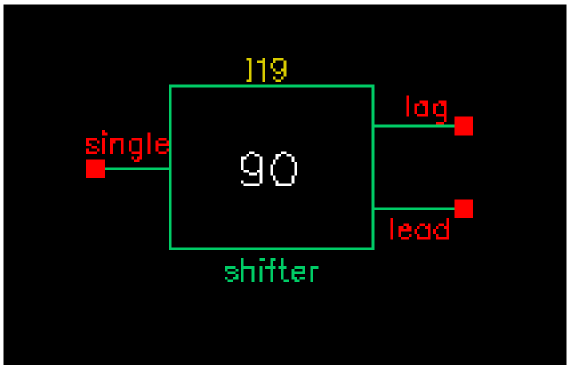
The rfVsource is a full RF signal generator that provides an 802.11 modulated RF signal for use in a circuit. The bit sequence is a bit short, so this source should not be used to certify the spectral content of your power amplifier. It should be used for a quick idea only.

Signal Standard	Modulation Type	Data Length	Step Period	Symbol Start	FFT Size	Prefix size	Frame Duration	Packet Length	Source EVM
802.11a	QAM-64	819.1u	12.5n	4u	64	16	4u	173u	.000151%
802.11g-ERP	BPSK	1m	12.5n		64	16	4u	One packet	.002313%
802.11 20M-Mix	QAM-64	400u	8.33333333n	0	64	8	3.6u	147.6u	.011109%
802.11 20M-Legacy	QAM-64	352u	8.33333333n	0	64	8	3.6u	131.6u	.02672%
802.11 20M-Greenfield	QAM-64	376u	8.33333333n	0	64	8	3.6u	139.6u	.014321%
802.11 40M-Mix	QAM-64	400u	8.33333333n	0	128	16	3.6u	90u	.105442%
802.11 40M-Legacy	QAM-64	320u	8.33333333n	0	128	16	3.6u	74u	.25561%
802.11 40M-Greenfield	QAM-64	360u	8.33333333n	0	128	16	3.6u	82u	.10558%
802.11 ac	QAM-64	448u	4.16666667n		256	64	4u	One packet	.000065%

© 1994-2020 All Rights Reserved.

shifter

(Phase Shifter)



In digital RF system designs, quadrature signal processing involves the phase splitting of high-frequency signals. The most common use of such components is to generate two signals that have a 90-degree phase difference based on one signal source (such as the RF signal or oscillator output). Another common use for a phase shifter is to combine two signals after adding a 90-degree phase difference, as in image-rejection receiver designs.

The Verilog-A module is declared as follows

```
module shifter(single, lag, lead);
    inout single, lag, lead;
    electrical single, lag, lead;
    parameter real freq = 1e9 from (0:inf);
    parameter real r    = 50 from (0:inf);
```

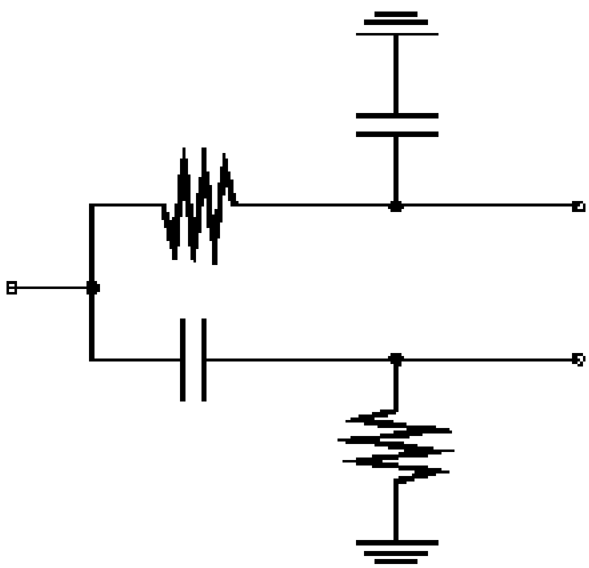
The parameters are:

`freq`	Frequency of operation [Hz].
`r`	Resistance [Ω] (see Figure 1-64).

Internally, the phase shifter is implemented using the RC-CR circuit as shown in Figure 1-64. While the phase difference is also 90-degrees when the `lead` and `lag` have the same output impedance, only at the operating frequency do the magnitudes remain the same. This circuit network also generates white noise.

© 1994-2020
All Rights Reserved.

Figure 1-64 Phase Shifter



There are two buffered versions of the shifter:

- The `shifter_combiner` combines two signals so that they add if one leads the other by 90 degrees and so that they cancel if it lags by 90 degrees.
- The `shifter_splitter` splits a signal into two signals 90 degrees out of phase with each other.
- You specify the input and output impedances. These networks are noiseless.

© 1994-2020
All Rights Reserved.

Triplexer

The `triplexer` is a series of three Bessel filters followed by three `portAdapters` and three ports. Bessel filters are chosen because there is no ripple in the passband of the filter and the transient response is not complicated. This allows a `tstab` to be set in the simulation to allow the filters to reach steady-state.

This is done to allow different reflection coefficients at the first three harmonics of the output frequency for a loadpull analysis on a power amplifier. Note that the output port is inside the triplexer schematic. To plot the loadpull, open-read the `triplexer` schematic and probe the terminal of the topmost port on the right side of the circuit.

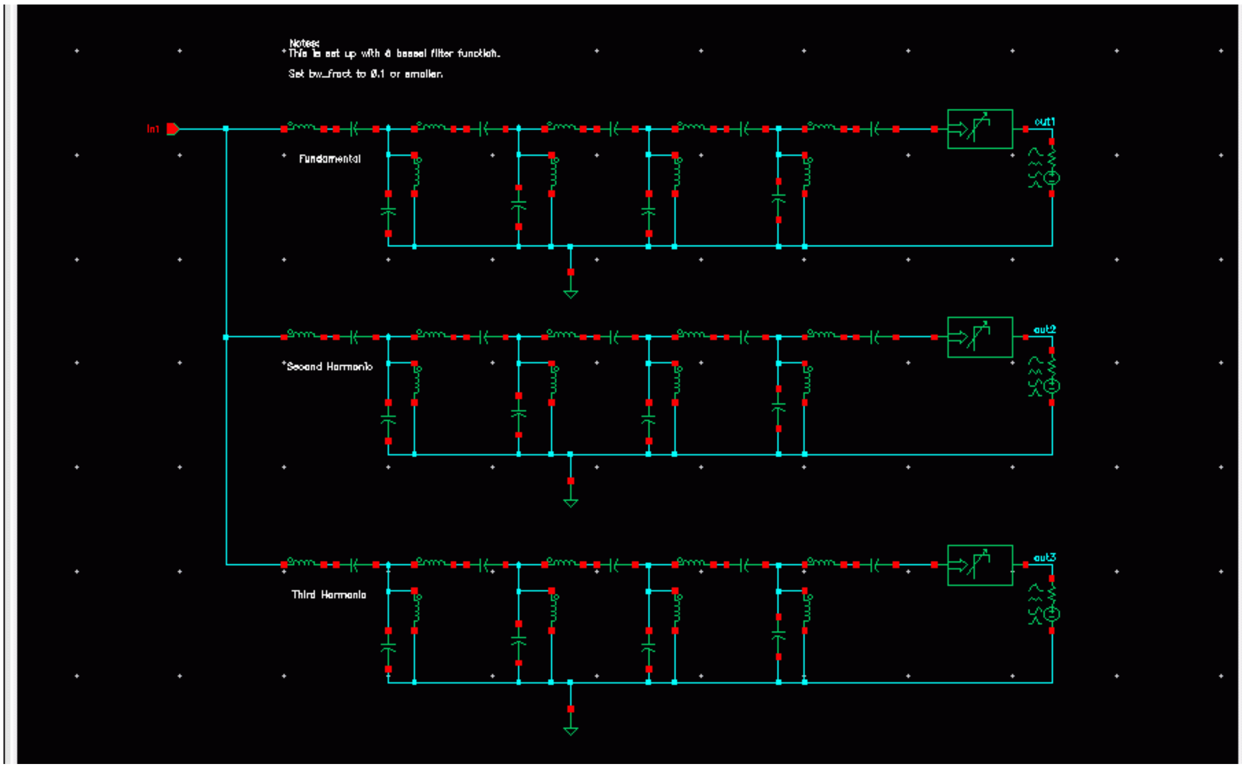
The bandpass filters must be kept quite narrow in order to minimize the interactions between the filters in the `triplexer` schematic. The bandwidth is set as a percentage of the frequency of the first filter.

The parameters are:

freq_harm1	This is the operating frequency of the amplifier.
Zo	This is the impedance of the system.
bw_fract	This is the fraction of the operating frequency for the amplifier. This should be 0.05 or smaller. 0.05 makes the bandwidth of all the filters 5% of the operating frequency.
mag_harm_1	This is the reflection coefficient for harmonic 1.
angle_harm_1	This is the angle of the reflection coefficient for harmonic 1.
mag_harm_2	This is the reflection coefficient for harmonic 2.
angle_harm_2	This is the angle of the reflection coefficient for harmonic 2.
mag_harm_3	This is the reflection coefficient for harmonic 3.
angle_harm_3	This is the angle of the reflection coefficient for harmonic 3.

© 1994-2020
All Rights Reserved.

The schematic of the `triplexer` is shown below.



If you wish to change the filter type, you can do so by copying the `triplexer` cell into a local library and editing the schematic. You can then edit the properties of the components. The first number of the numerator or denominator is the one radian per second and 1 ohm value you can find in any filter design manual. If you have a choice, put the largest values in the leftmost component of the filter. When the value for the leftmost inductor is less than the default of 2.2649, the bw_fract should be reduced in order to prevent interaction between the filters.

© 1994-2020 All Rights Reserved.

Ten_plexer

The `ten_plexer` is a series of ten Bessel filters followed by ten `portAdapters` and ten ports. Bessel filters are chosen because there is no ripple in the passband of the filter and the transient response is not complicated. This allows a `tstab` to be set in the simulation to allow the filters to reach steady-state.

This is done to allow different reflection coefficients at the first ten harmonics of the output frequency for a loadpull analysis on a power amplifier. Note that the output port is inside the ten_plexer schematic. To plot the loadpull, open-read the ten_plexer schematic and probe the terminal of the topmost port on the right side of the circuit.

The bandpass filters must be kept quite narrow in order to minimize the interactions between the filters in the ten_plexer schematic. The bandwidth is set as a percentage of the frequency of the first filter.

The parameters are:

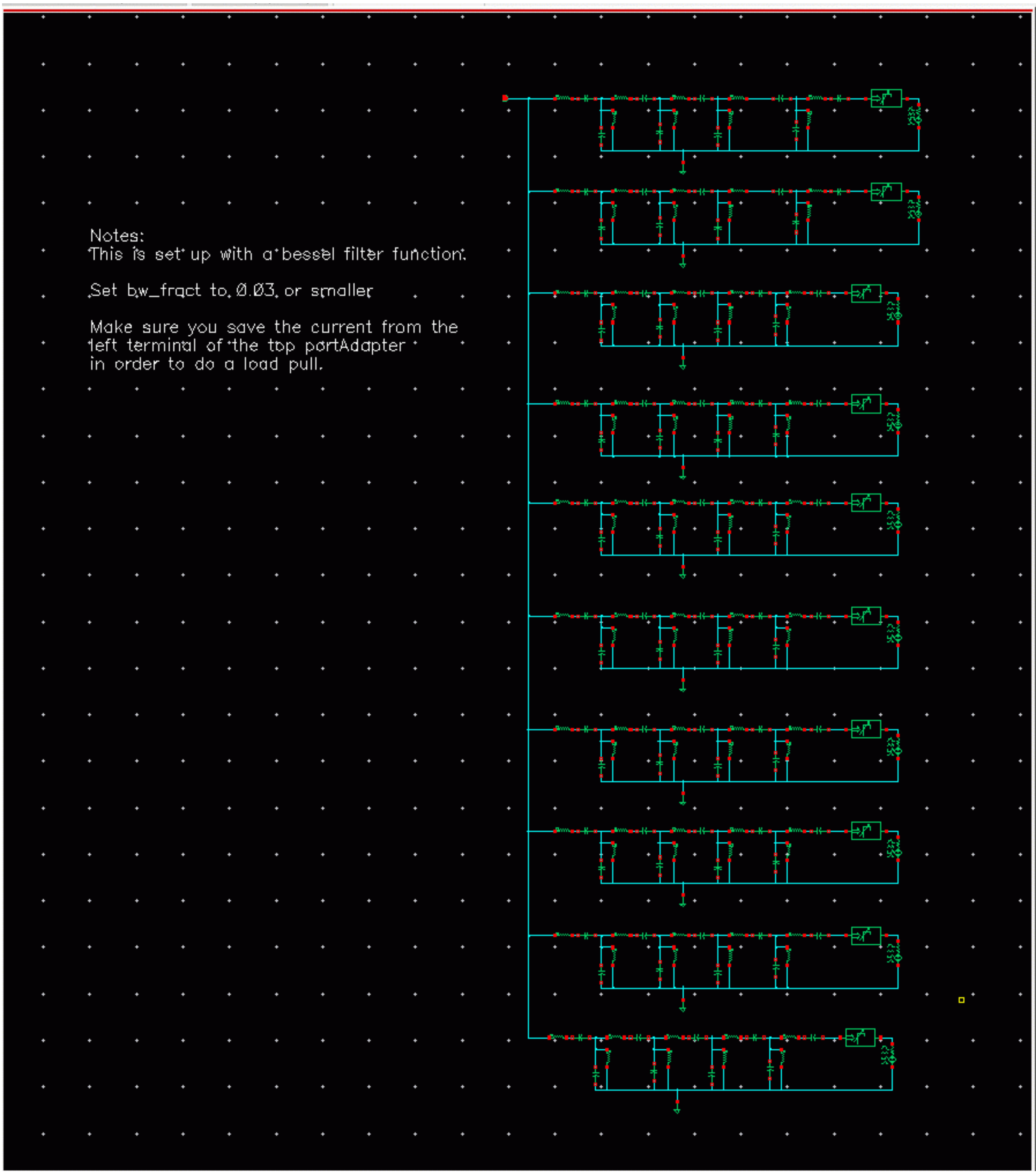
`freq_harm1`	This is the operating frequency of the amplifier.
`Zo`	This is the impedance of the system.
`bw_fract`	This is the fraction of the operating frequency for the amplifier. This should be 0.03 or smaller. 0.03 makes the bandwidth of all the filters 3% of the operating frequency.
`mag_harm_1`	This is the reflection coefficient for harmonic 1.
`angle_harm_1`	This is the angle of the reflection coefficient for harmonic 1.
`mag_harm_2`	This is the reflection coefficient for harmonic 2.
`angle_harm_2`	This is the angle of the reflection coefficient for harmonic 2.
`mag_harm_3`	This is the reflection coefficient for harmonic 3.
`angle_harm_3`	This is the angle of the reflection coefficient for harmonic 3.

© 1994-2020 All Rights Reserved.

`mag_harm_4`	This is the reflection coefficient for harmonic 4.
`angle_harm_4`	This is the angle of the reflection coefficient for harmonic 4.
`mag_harm_5`	This is the reflection coefficient for harmonic 5.
`angle_harm_5`	This is the angle of the reflection coefficient for harmonic 5.
`mag_harm_6`	This is the reflection coefficient for harmonic 6.
`angle_harm_6`	This is the angle of the reflection coefficient for harmonic 6.
`mag_harm_7`	This is the reflection coefficient for harmonic 7.
`angle_harm_7`	This is the angle of the reflection coefficient for harmonic 7.
`mag_harm_8`	This is the reflection coefficient for harmonic 8.
`angle_harm_8`	This is the angle of the reflection coefficient for harmonic 8.
`mag_harm_9`	This is the reflection coefficient for harmonic 9.
`angle_harm_9`	This is the angle of the reflection coefficient for harmonic 9.
`mag_harm_10`	This is the reflection coefficient for harmonic 10.
`angle_harm_10`	This is the angle of the reflection coefficient for harmonic 10.

© 1994-2020 All Rights Reserved.

The schematic of the `ten_plexer` is shown below.



If you desire to change the filter type, you can do so by copying the `ten_plexer` cell into a local library, and editing the schematic. You can then edit the properties of the components.

© 1994-2020
All Rights Reserved.

The first number of the numerator or denominator is the one radian per second and 1 ohm value you can find in any filter design manual. If you have a choice, put the largest values in the leftmost component of the filter. When the value for the leftmost inductor is less than the default of 2.2649, the `bw_fract` should be reduced in order to prevent interaction between the filters.

© 1994-2020 All Rights Reserved.

Testbenches Category

The `testbenches` category contains the test circuits used to define model specifications. Where possible, the element names are 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 this purpose.

The components in the `testbenches` category are:

- AM_PM_test_ckt
- ava_pwr_gain
- BB_ind_cap_test
- demod_ip3
- dwn_cnvt_test
- mixer_ip3
- mod_1dbcp
- mod_demod_test
- noise_figure
- one_db_cp
- PB_BB_filter_comparison
- PB_ind_cap_test
- quad_and_phase_error_demo
- shifter_combiner_test
- shifter_splitter_test
- up_cnvt_test
- view_switching

© 1994-2020 All Rights Reserved.

AM_PM_test_ckt

(AM/PM Conversion Parameters)

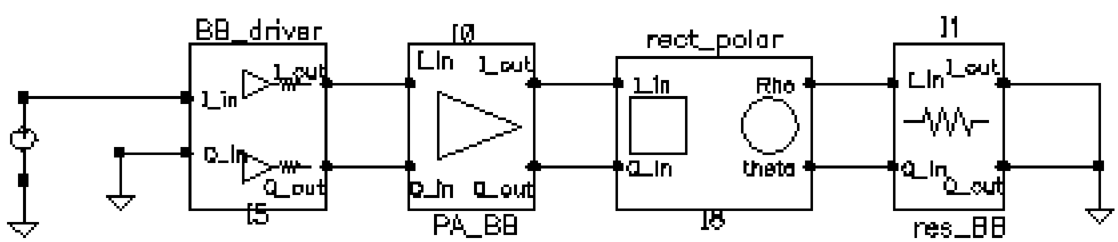
Only the baseband models include the four parameters for AM/PM conversion.

Table 1-1 AM/PM Conversion Parameters for Baseband Models

AM/PM Parameter	Definition
`AM/PM Sharpness`	Defines how steep the output phase shift changes are with respect to input power.
`{1, 0, -1}` for `{cw, none, ccw}`	Defines the direction of the phase shift. 1 for clockwise, 0 for no phase shift, -1 for counter clockwise.
`|radians|@1dB cp`	Defines the absolute value of the output phase shift at the 1dB compression point for power amplifiers. This is the phase shift at an arbitrary output power level for some models.
`|radians|@big input`	Defines the absolute value of the output phase shift as input power goes to infinity (if it could go to infinity).

The test circuit in Figure 1-65 is listed as `am_pm_test_ckt` in the `testbenches` category in `rfLib`.

Figure 1-65 The am_pm_test_ckt Circuit



In the `am_pm_test_ckt` test circuit,

- The first block (`BB_driver`) scales the control voltage generated by the leftmost element so that the output equals the specified dBm when the control voltage equals 1

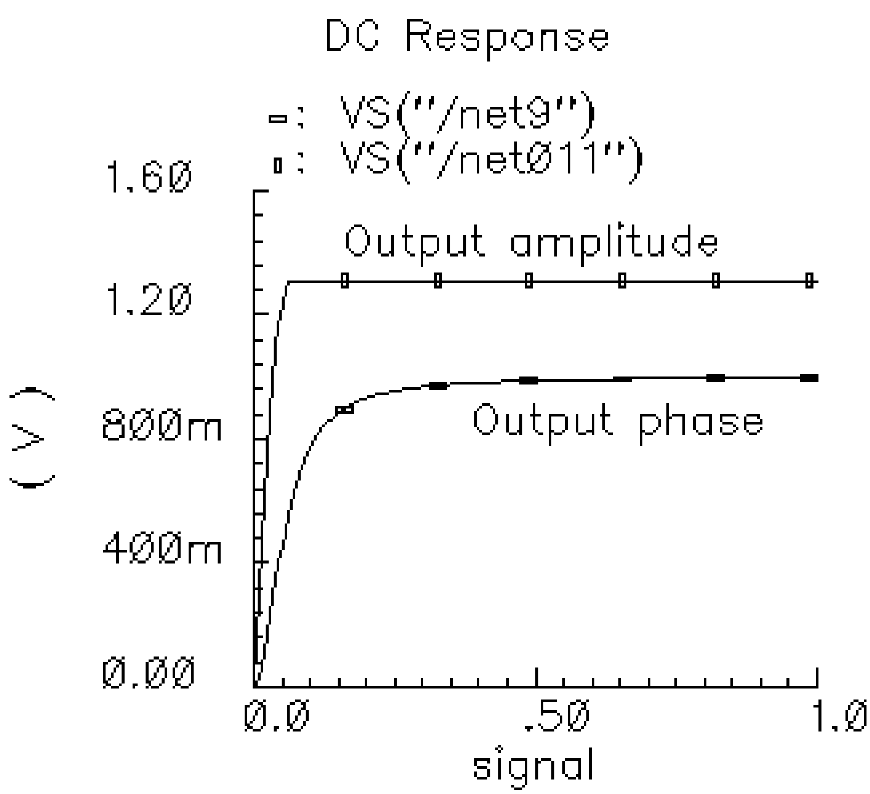
© 1994-2020 All Rights Reserved.

volt. This is done so you can specify maximum dBm but still sweep linearly from zero signal.

- The second block (`PA_BB`) is a power amplifier.
- The third block (`rect_polar`) transforms the rectangular description of the baseband signal into polar coordinates so you can observe the phase shift and output signal level directly.

Figure 1-66 shows the output amplitude and phase as functions of the input signal level. Generate these with a swept DC analysis. Sweep the *signal* variable from 0 to 1 in 200 linear steps and display the *rect_polar* outputs.

Figure 1-66 Output Amplitude and Phase

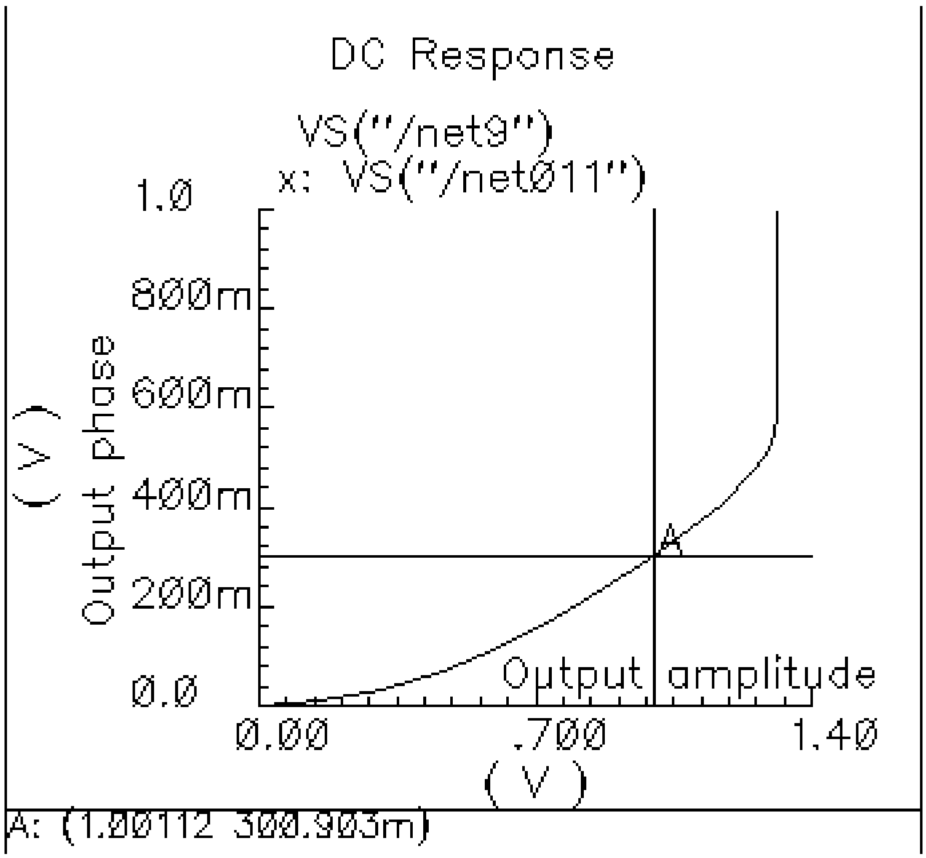


By changing the x-axis to be the output amplitude trace, you can confirm that the phase shift at the output referred 1 dB compression point of 10dBm (or 1 volt peak across a 50 ohm load) equals 0.3 radians, as specified. Figure 1-67 shows the plot.

© 1994-2020
All Rights Reserved.

Note that the measured power across the load is as specified only when the load matches the amplifier output resistance. If you mismatch the load you do not measure the specified phase shift at the specified output power level.

Figure 1-67 Output Phase Versus Output Amplitude



In the next three figures, output phase is plotted against input signal level. Each plot shows the effect of one of the AM/PM conversion parameters. You can generate the plots by applying the Parametric Tool to the existing analysis.

Figure 1-68 shows the effect of the `|radians|@1 db cp` parameter. Sweep `rad_cp` from 10 m to 100 m in 5 linear steps.

© 1994-2020 All Rights Reserved.

Figure 1-68 Output Modified by the |radians|@1 db cp Parameter

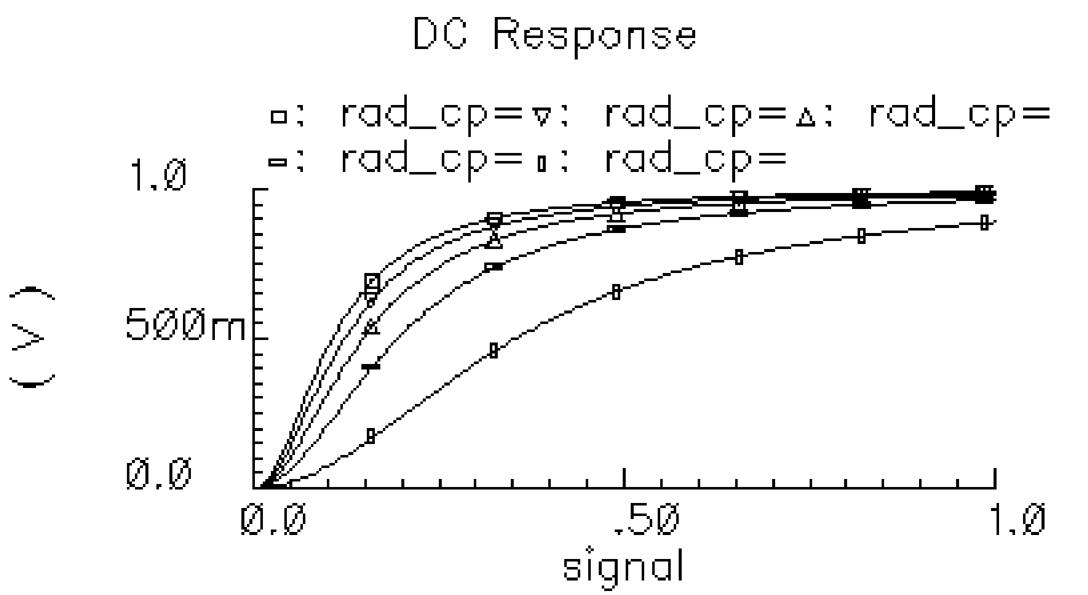
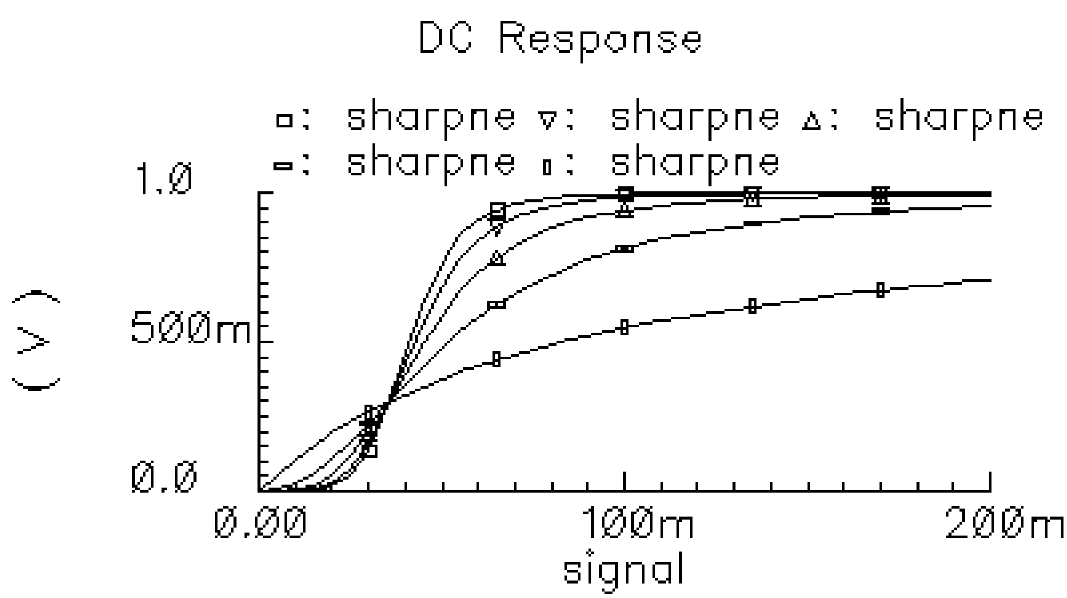


Figure 1-69 shows the effect of the `am/pm sharpness` parameter. Sweep `sharpness` from 1 to 6 in 5 linear steps.

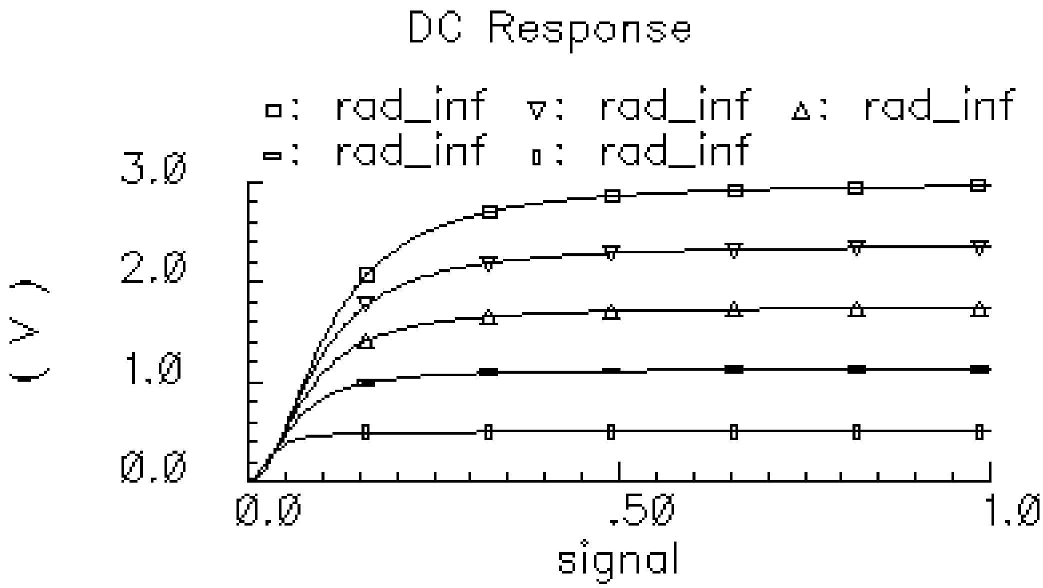
Figure 1-69 Output Modified by the Sharpness Parameter



© 1994-2020 All Rights Reserved.

Figure 1-70 shows the effect of the `rad_inf` parameter. Sweep `rad_inf` from 0.5 to 3 in 5 linear steps.

Figure 1-70 Output Modified by the rad_inf Parameter



© 1994-2020
All Rights Reserved.

ava_pwr_gain

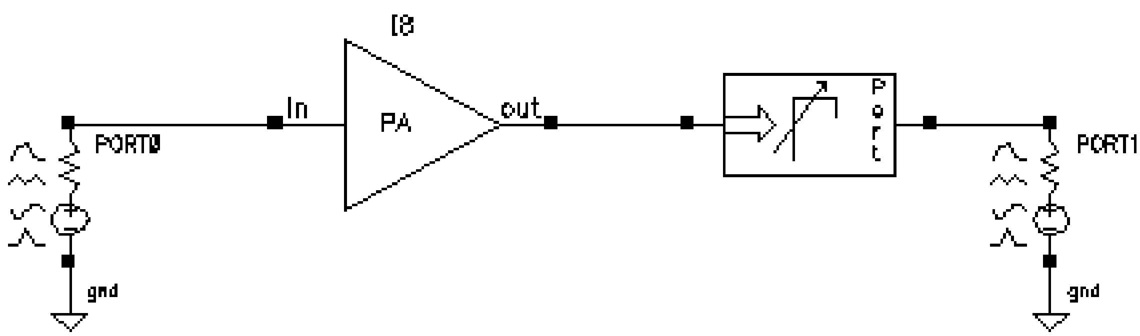
(Available Power Gain Parameter)

When an amplifier's load is equal to it's output resistance, available power gain equals the following

$$10 \times \log\left(\frac{outputpower}{inputpower}\right)$$

The test circuit in Figure 1-71 is listed as `ava_pwr_gain` in the `testbenches` category in `rfLib`.

Figure 1-71 The ava_pwr_gain Circuit



Computing Constant Power Contours

The `ava_pwr_gain` test circuit is set up to compute constant power contours. As you would expect, maximum power transfer occurs when the load and output impedances are matched. The port adapter inserts reactive elements into the signal path to load the amplifier with the specified reflection coefficient.

Figure 1-72 on page 166 shows a Smith Chart that displays how the load power varies with the load refection coefficient.

The load pull contours were computed by

- Sweeping the `pp` parameter in a PSS analysis (`pp` is the *phase* of the reflection coefficient)

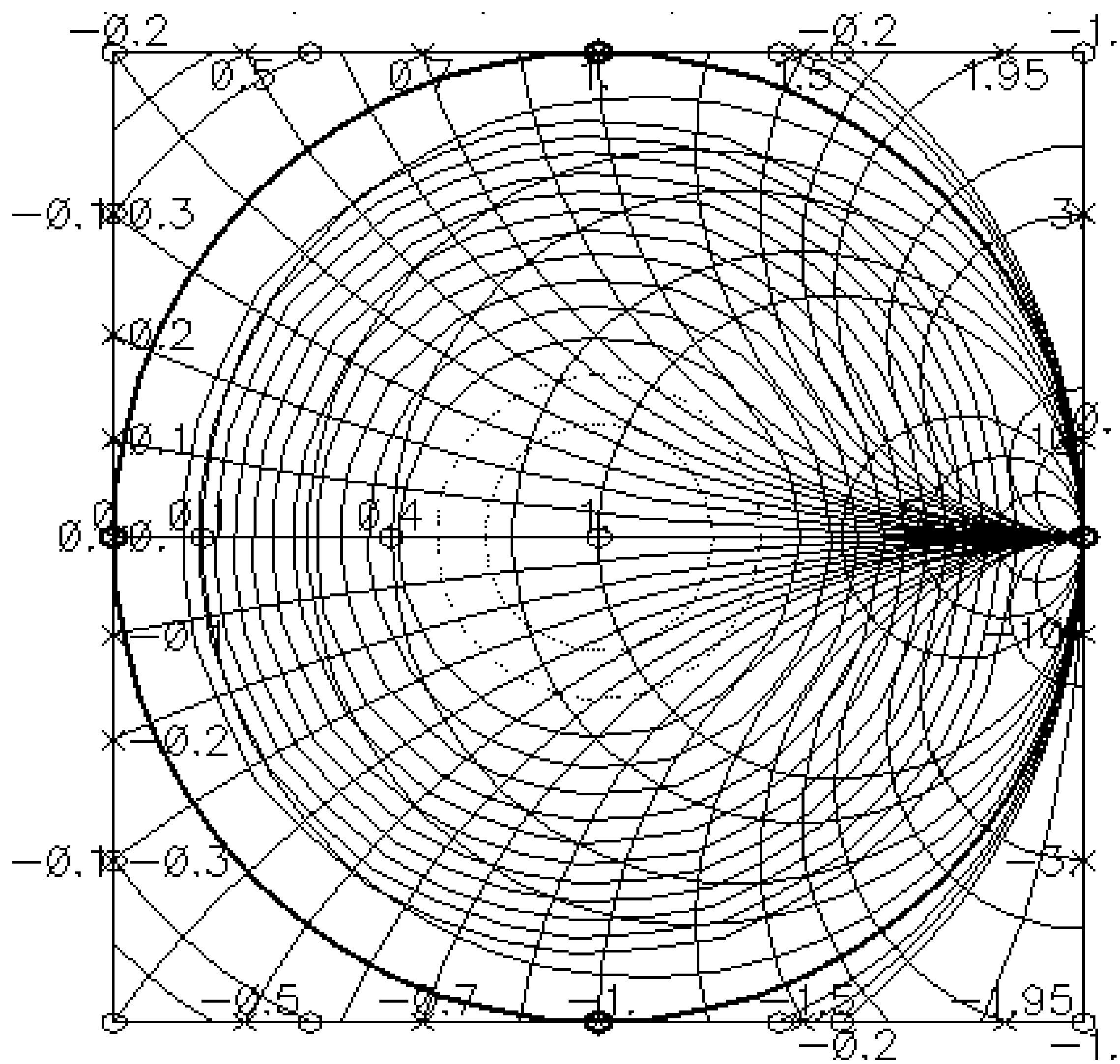
© 1994-2020
All Rights Reserved.

- Sweeping the `mm` parameter with the Parametric Tool (`mm` is the *magnitude* of the reflection coefficient)

The *load reflection coefficient* is defined with reference to the amplifier output resistance, `300` Ohms in this case. The amplifier input resistance is `20` Ohms. The input source resistance is `50` hms. The amplifier 1 dB compression point is set high enough to make the amplifier linear. The available power gain parameter is `20` dB.

To generate the load pull contours you must save *both* the current flowing into the port adapter (`port`) and the current flowing into `Port0`.

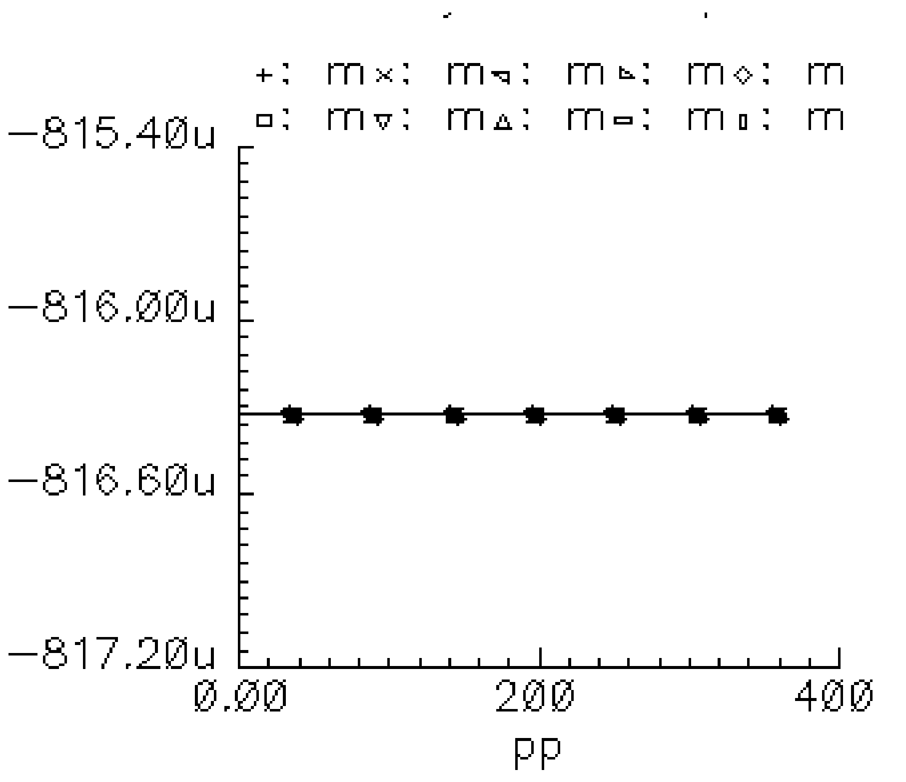
Figure 1-72 Smith Chart



© 1994-2020
All Rights Reserved.

When you place the cursor on the smallest contour on the Smith Chart, you can see that the amplifier delivers a maximum power of `81.63` mW to an optimum load of `300` Ohms (reflection coefficient = 0). When you plot the magnitude of the power coming from the input port against the sweep variable (`pp`, phase of the reflection coefficient) you find that input power equals `816.3` uW, independent of load, as shown in Figure 1-73. The ratio of maximum output to input power equals `100`, or dB, as specified.

Figure 1-73 Input power



Note that the voltage gain in this test circuit does not equal 10 because the amplifier's input and output resistances are different. You can verify that the ratio of the output to input voltage is as follows

$$10\sqrt{\frac{R_{out}}{R_{in}}}$$

where, R_{out} is the amplifier output resistance and R_{in} is the amplifier input resistance. This assumes the amplifier is not driven into non-linear operation.

The input and output resistances specify the current drawn by the associated terminals as a linear function of terminal voltage. There is no test circuit for terminal resistances because the definition is so simple.

© 1994-2020 All Rights Reserved.

BB_ind_cap_test

(RLC Test Circuits)

The two circuits discussed below demonstrate how passband and baseband reactive elements are related. The circuit in Figure 1-74 shows a simple passband RLC circuit driven by a modulated carrier. The circuit in Figure 1-75 shows the associated baseband equivalent circuit model. The circuits are `PB_ind_cap_test` and `BB_ind_cap_test`. Both circuits reside in the `rfLib` under the `testbenches` category.

Figure 1-74 Simple Passband RLC Circuit

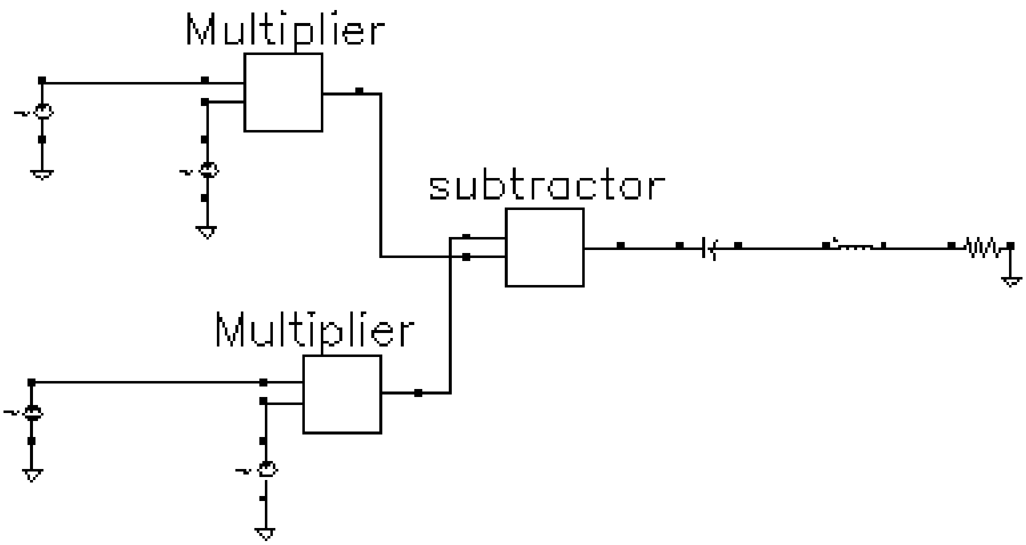
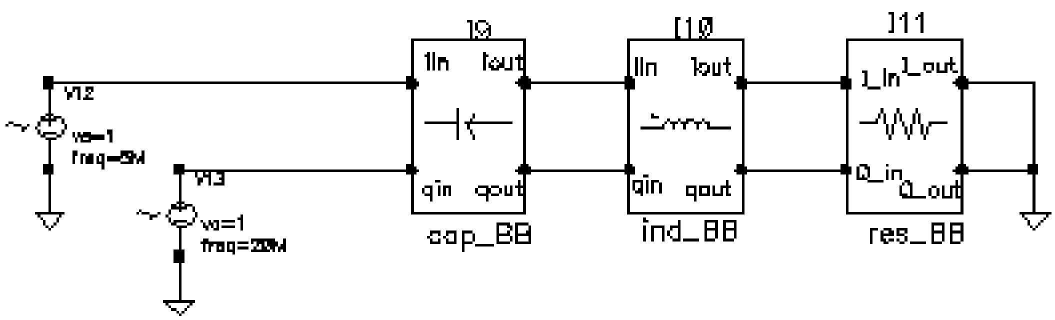


Figure 1-75 Baseband Equivalent To Figure 1-63



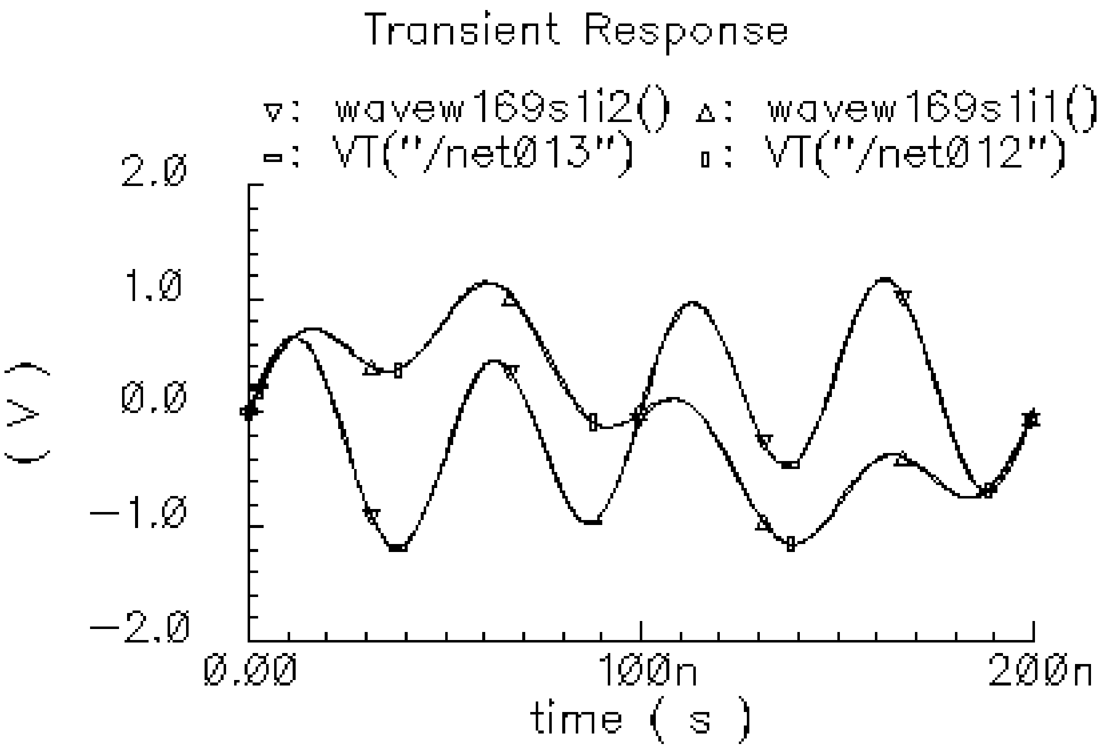
The following steps explain how to simulate each circuit and overlay the results.

© 1994-2020
All Rights Reserved.

1. Recall the `PB_ind_cap_test` circuit and bring up an analog design environment window. Set up a 200 ns Envelope analysis. Select `carrier` as the *Clock Name*. Set the *Output Harmonics* to 1.
2. Run the analysis and plot the real and imaginary parts of the *harmonic-time voltage* across the resistor. Use `1` for the harmonic number.
3. Recall the `BB_ind_cap_test` circuit and run a 200 ns transient analysis. Note the faster run time. That is the whole point to suppressing the carrier but it is only useful if the results match. Plot the `I_in` and `Q_in` voltages of the resistor model.
4. To overlay the results, bring up a waveform calculator.
5. Click the *wave* button on the calculator then click one of the Envelope waveforms. If the waveform turns yellow you may have to hit the escape button a few times and click *clear* and *clst* a couple of times in the calculator then try again.
6. Make active the waveform display tool with the transient results then click *Plot* in the calculator.
7. Repeat the last two steps for the other Envelope waveform. You should see the waveforms in Figure 1-76. The two models agree very well. The resonant frequency of the series RLC branch is just over 500 MHz. Only by riding on a carrier can the 5 MHz and 20 MHz baseband signals propagate to the resistor at their original voltage levels. The baseband model accurately predicts the effects of the RLC circuit on the baseband signal. There are two effects, one due to phase shift at the carrier frequency and one due to filtering of the baseband signal itself.
8. In the waveform display tool that overlays the results, change the x-axis to be one of the I-signals. You should get the picture shown in Figure 1-77. The tilt in the resulting Lissajous plot indicates phase shift at the carrier frequency but not at the baseband frequencies. The aspect ratio of the Lissajous figure indicates the 20 MHz component is attenuated more than the 5 MHz component. The baseband model captures both effects well.

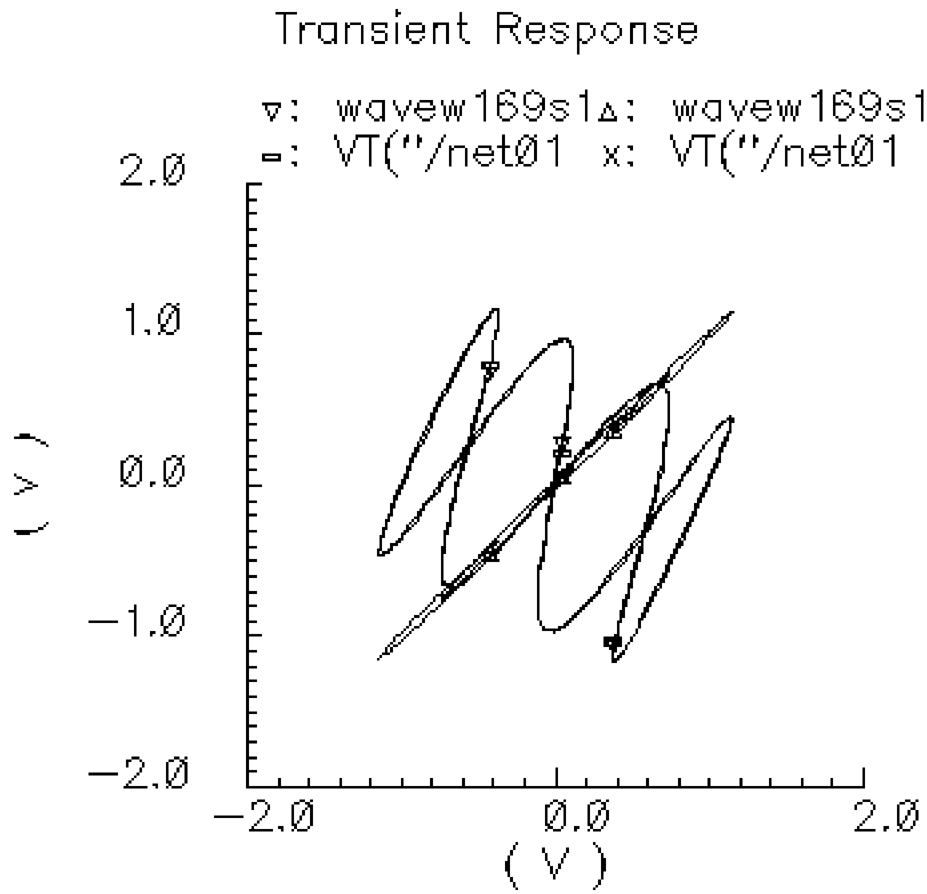
© 1994-2020 All Rights Reserved.

Figure 1-76 Waveforms



All Rights Reserved.

Figure 1-77 Lissajous plot

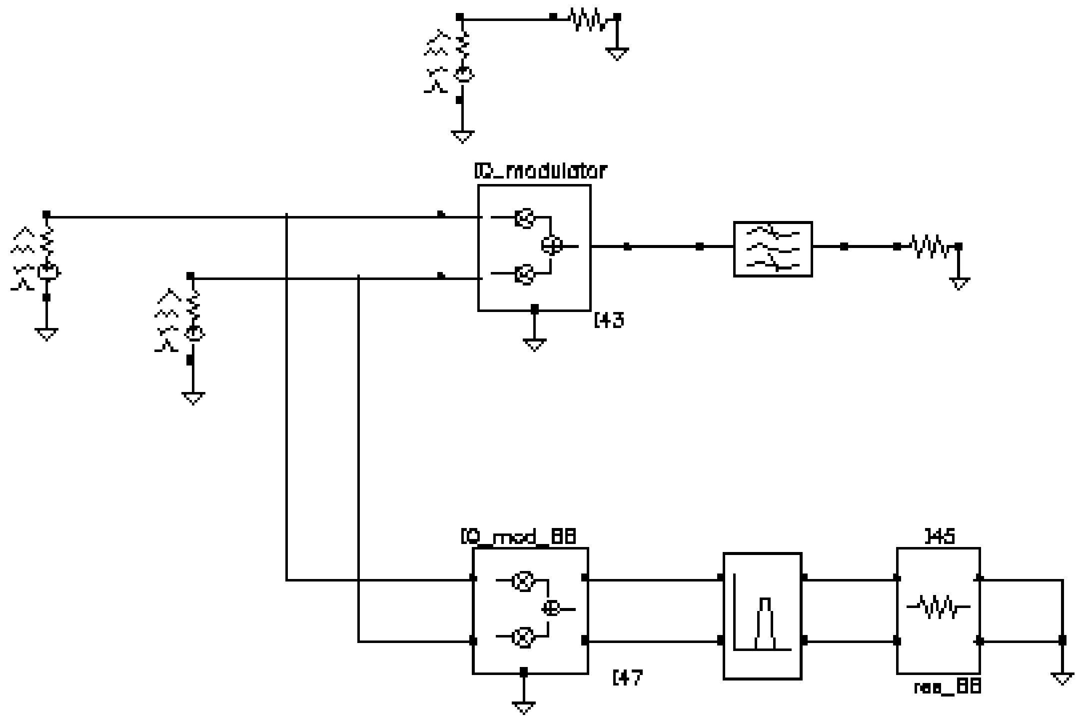


Comparison of Baseband and Passband Models

The circuit in Figure 1-78 on page 173 shows how well the baseband and passband filters agree. The I-input is a 5MHz 1 volt peak sinusoid and the Q-input signal is a 20MHz 1 volt peak signal. The filter has a center frequency of 1.1GHz and a relative bandwidth of 0.1. The modulator LO is 1GHz. To make the analysis more interesting the carrier is not exactly aligned with the filter center frequency and the terminals are not matched. The circuit is listed as `PB_BB_filter_comparison` in the testbenches category of the `rfLib`.

© 1994-2020 All Rights Reserved.

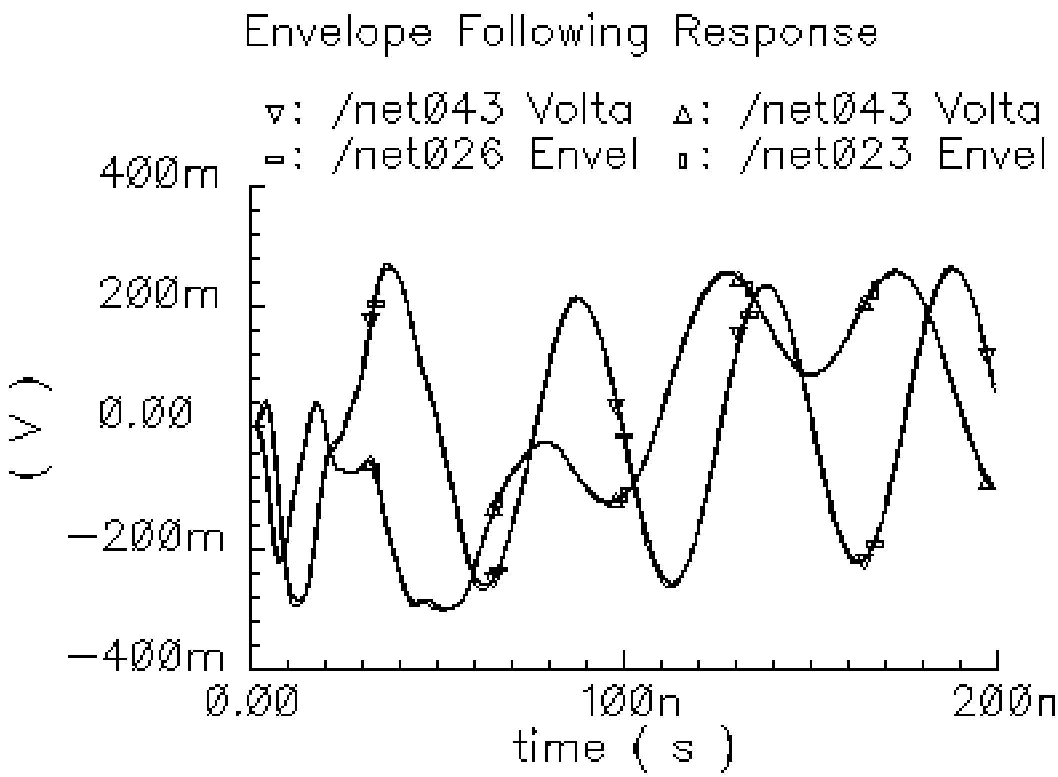
Figure 1-78 PB_BB_filter_comparison Circuit



1. Bring up the test circuit and an Analog Environment window.
2. Set up an Envelope analysis with “carrier” as the Clock Name. Set reltol in the analog options to 1e-5. You can use the default reltol of 1e-3 but you do not get the waveforms close to the baseband results.
3. Plot the “time” waveforms of the BB_butterworth_bp outputs. These waveforms are the response of the baseband equivalent model.
4. Plot the “harmonic time”, 1 harmonic, real and imaginary waveforms at the butterworth_lp output. These waveforms are the baseband waveforms extracted from a passband model. Figure 1-79 on page 174 overlays the baseband and passband results. The baseband and passband filter models produce identical equivalent baseband waveforms. The slight offset in time is due to the ambiguity associated with deciding whether to plot a time-varying Fourier coefficient at the beginning or at the end of a clock cycle.

© 1994-2020 All Rights Reserved.

Figure 1-79 I and Q Baseband Equivalent Outputs



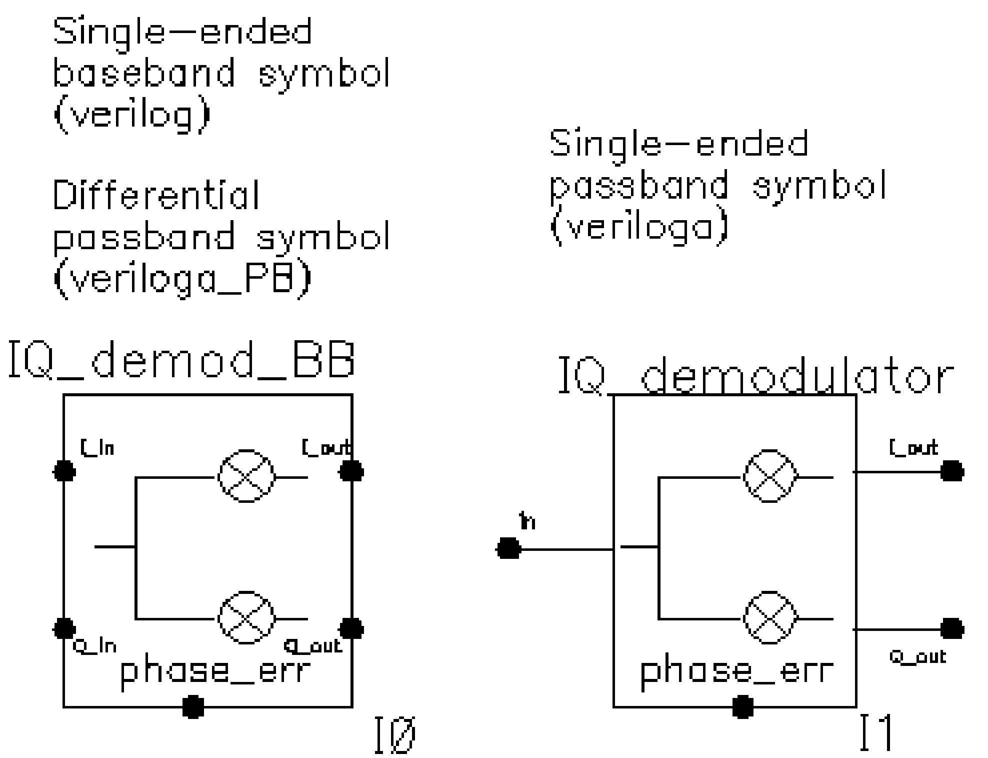
© 1994-2020
All Rights Reserved.

demod_ip3

(IQ Demodulator)

`(baseband = IQ_demod_BB, passband = IQ_demodulator)`

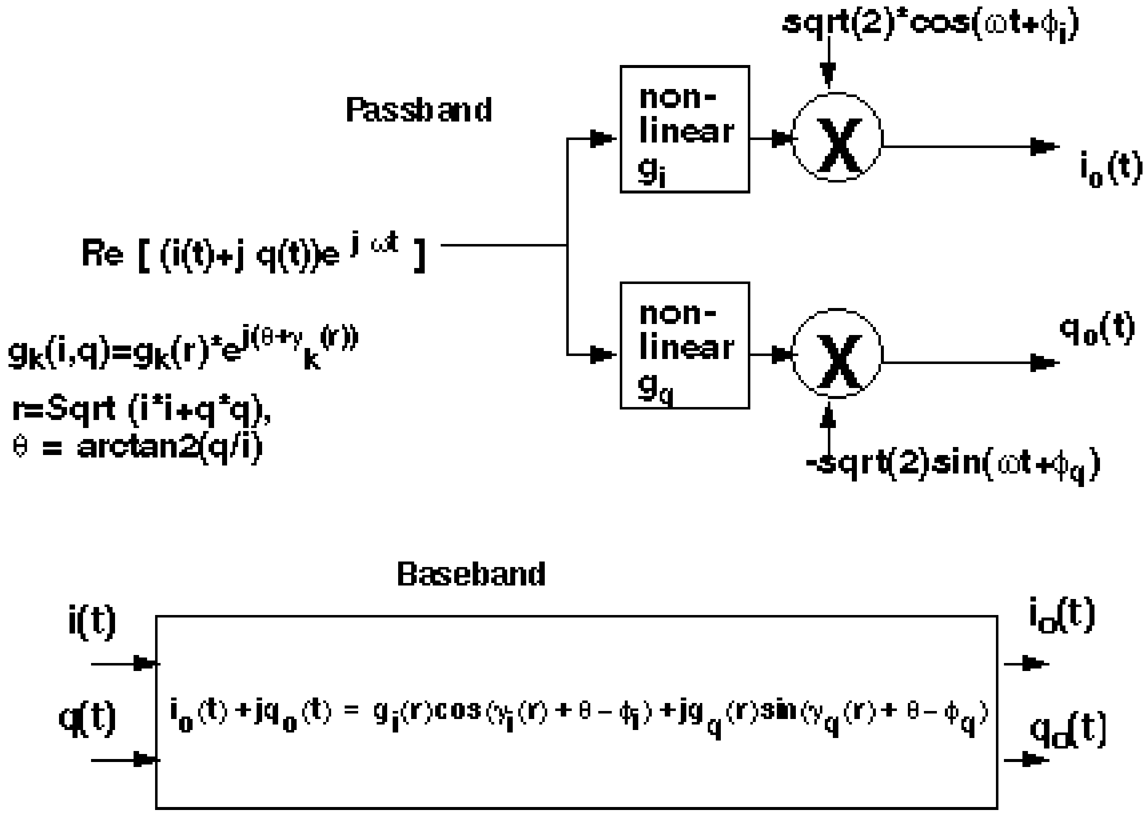
Figure 1-80 Baseband and Passband IQ Demodulator Models



The `IQ_demodulator` converts RF (or IF) to baseband. Figure 1-81 on page 176 shows exactly what the passband demodulator model does. The parameters are like those in the modulator blocks except saturation is specified by input referred IP3 instead of by 1 dB compression point. IP3 was chosen over the 1 dB compression point for specifying saturation because the demodulator usually lies in the receive path and receiver blocks are usually specified with IP3.

© 1994-2020 All Rights Reserved.

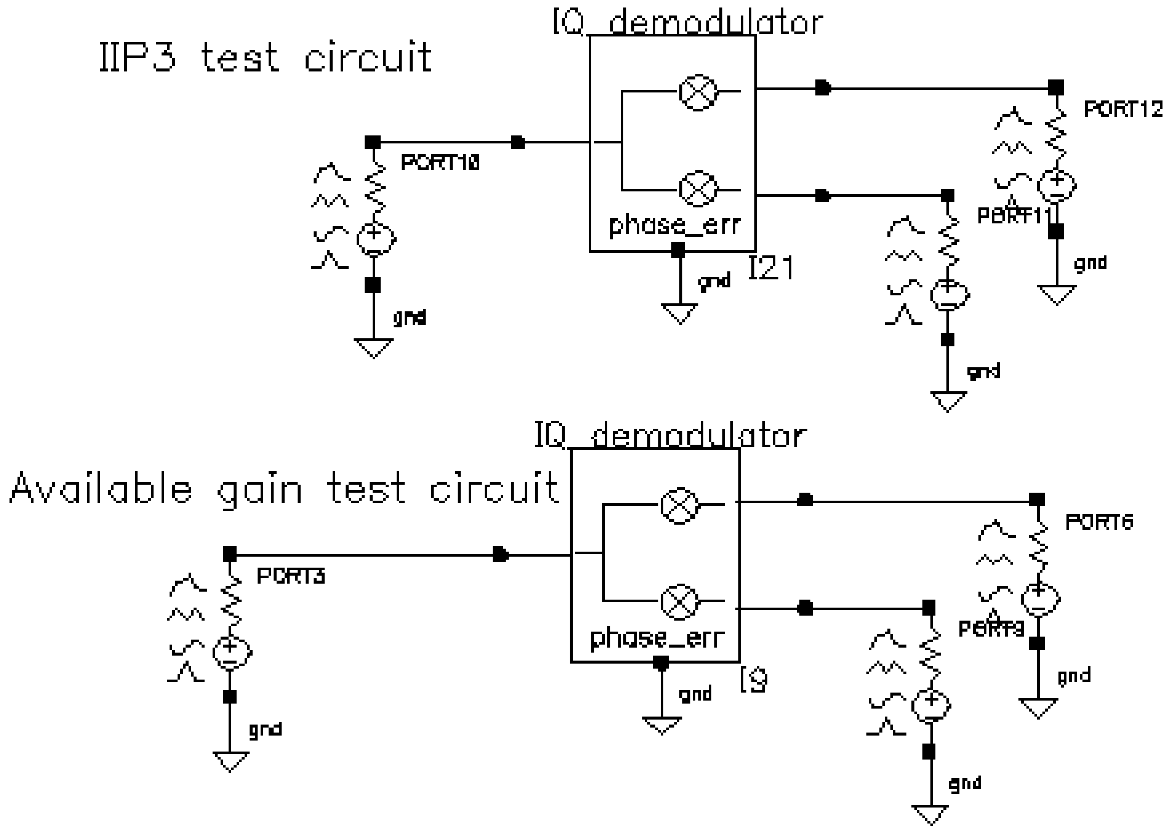
Figure 1-81 IQ Demodulator Calculations



The circuit called `demod_ip3` in the `testbenches` category of the `rfLib` shows how the gain and IP3 parameters are defined. Figure 1-82 shows the schematic. Both the input and the output resistances are matched.

© 1994-2020 All Rights Reserved.

Figure 1-82 The demod_IP3 Schematic



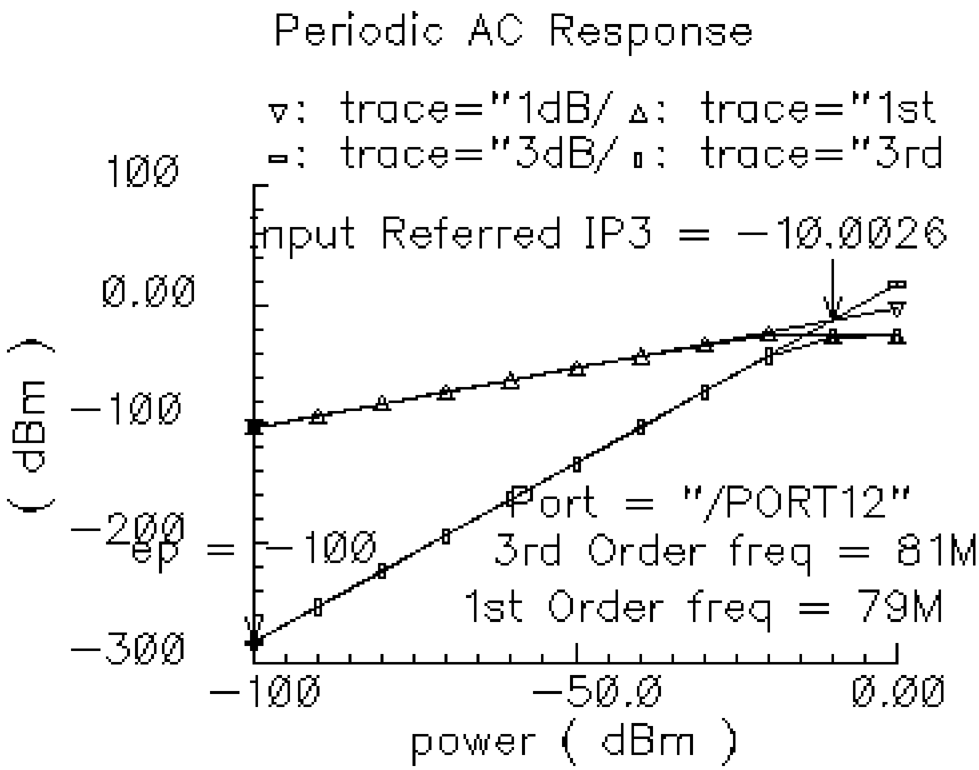
1. Recall the `demod_IP3` circuit and set up a swept PSS analysis. Let the *Beat Frequency* be *Auto Calculated*. Keep 2 harmonics. Sweep the *power* parameter from -100 to 0 in 10 linear steps.

2. Set up a single point PAC analysis at 921 MHz and keep the -25 and -21 sidebands.

3. After running the analysis, from the PAC output window plot the input-referred IP3 curves with 81 MHz as the 3rd order sideband and 79 MHz as the 1st order sideband. The procedure is similar to the mixer IP3 example covered in "(IQ Modulator Models)" on page 36. Use *Variable Sweep* for the *Circuit Input Power* and -100 for the *Extrapolation point*. Make sure to plot Input Referred IP3. Click the `I-output` port in the top circuit. You should see -10 dBm as the IP3, just as specified. Figure 1-83 shows the IP3 plot. Note that 1st order line indicates the gain is 3dB below the specified gain of 0 dB. That is because not all of the power lies at 1000 MHz-921 MHz = 79 MHz; Some of the power lies at 1000 MHz + 921 MHz = 1921 MHz. Use the bottom test circuit to measure available power gain. The bottom circuit drives the demodulator at the same frequency as the demodulator's internal local oscillator, which runs at 1 GHz. Now the output power is not split, it lies in the zero harmonic of the `I-output`.

© 1994-2020 All Rights Reserved.

Figure 1-83 Demodulator IP3

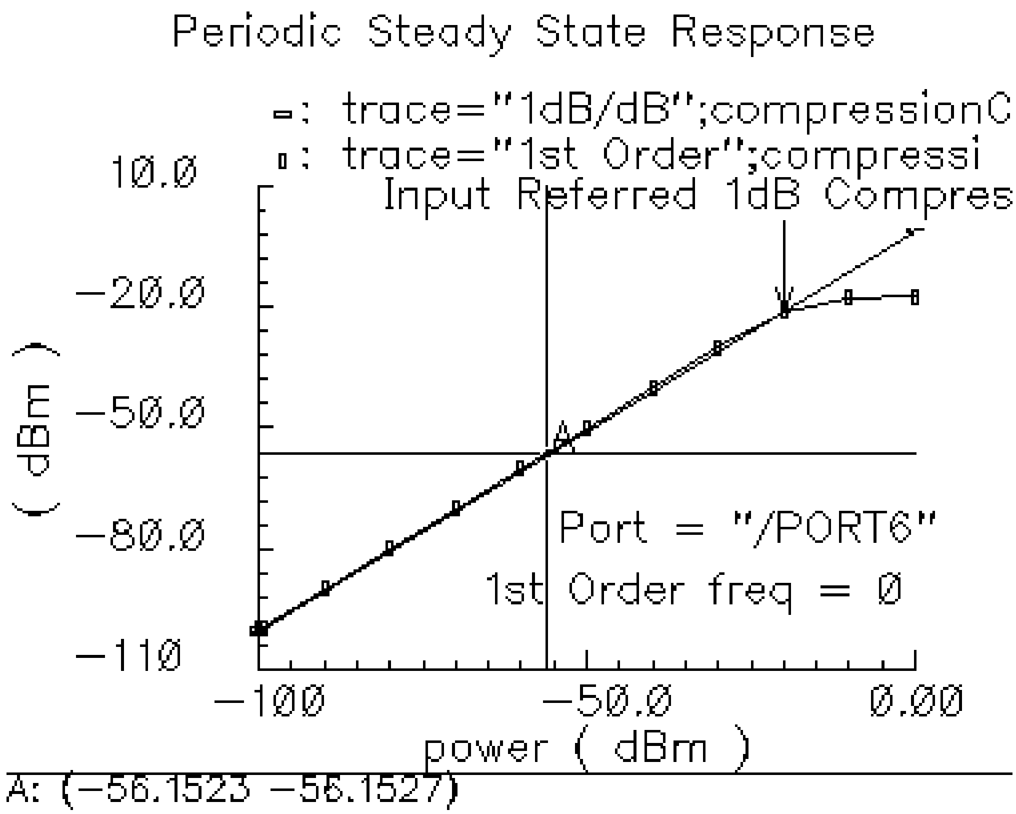


4. Plot the 1dB compression point at the port loading the `I-output` of the bottom circuit. Use the zeroth harmonic. The ratio of output to input power should be unity in the linear region. Figure 1-84 shows the compression point plot. The measured 1dB compression point is of no use in this test. We want the gain. At low power levels where the gain is constant, the gain is as specified.

5. Remember, in this test circuit the load resistance and output resistance are equal so that the output power is maximal. Also, the input resistance equals the source resistance so that the horizontal axis truly equals input power.

© 1994-2020 All Rights Reserved.

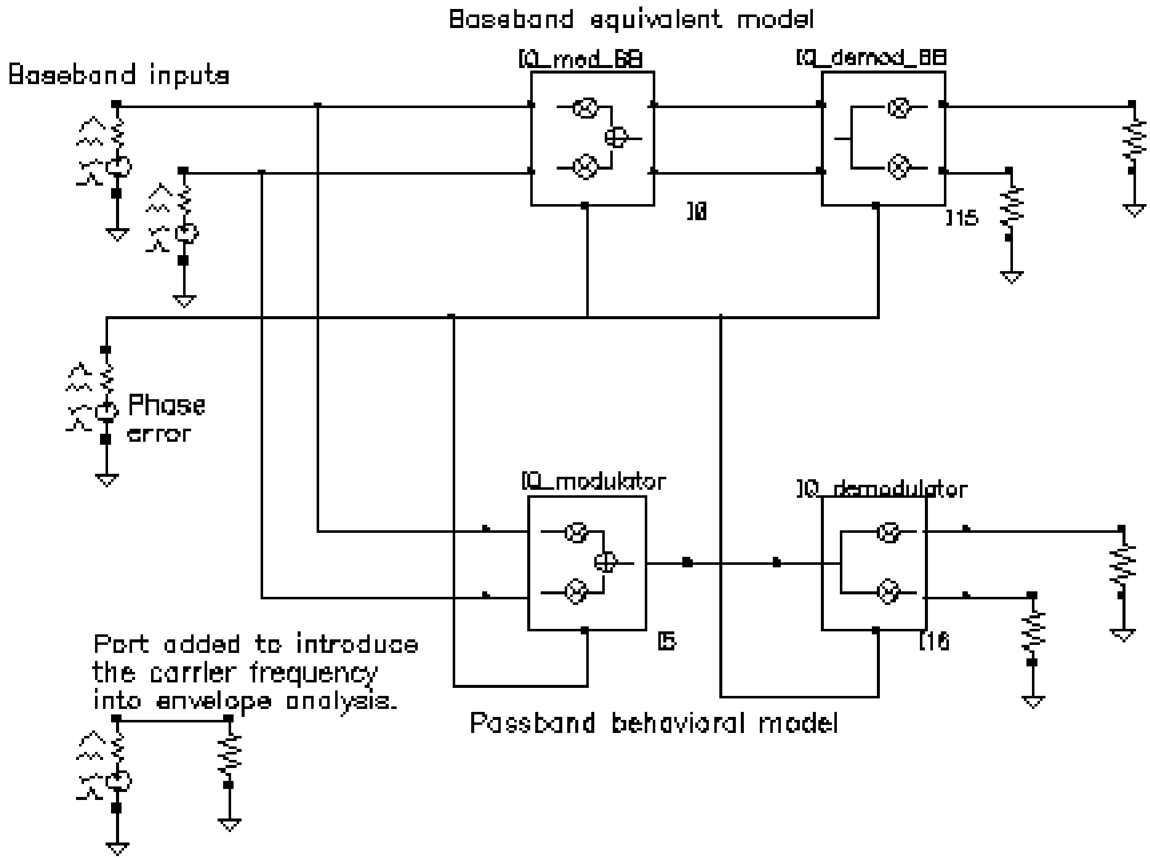
Figure 1-84 Demodulator Available Power Gain



Phase errors behave like their counterparts in the modulator models except for a change of sign. Quadrature error behaves exactly as it does in the modulator models. Figure 1-85 shows a test circuit for illustrating the relationships between phase error and quadrature error in the modulators and demodulators. The test circuit is called `mod_demod_test` and is listed in the `testbenches` category. The test circuit also shows that the passband and baseband models give comparable results, as they should, as long as the passband carrier is not severely clipped. The baseband input trajectory is a complex 1 MHz tone, which produces a circular input trajectory. The demodulator outputs are not matched and are not symmetric with respect to I and Q paths. The modulators and demodulators are not perfectly linear and the non-linearities are asymmetric with respect to I and Q. The modulators and demodulators are driven by the same phase error and the quadrature error parameters are a common variable set to 0.785 radians.

© 1994-2020
All Rights Reserved.

Figure 1-85 mod_demod_test Circuit



To use the `mod_demod_test` circuit:

1. Recall the circuit and set up a 5 us Envelope analysis with `carrier` as the *Clock Name*.
2. After the analysis completes, plot the `IQ_mod_BB` outputs and make the `I_out` signal the x-axis.
3. Open a subwindow and in it, plot the harmonic time waveforms of the `IQ_modulator` output. Use the first harmonic and plot the real and imaginary waveforms. Make the real waveform the x-axis.
4. Open a third subwindow and stretch the Waveform Display window so that the third subwindow appears below the first window.
5. Plot the time waveforms at the `IQ_demod_BB` outputs and make the `I_out` waveform the x-axis.

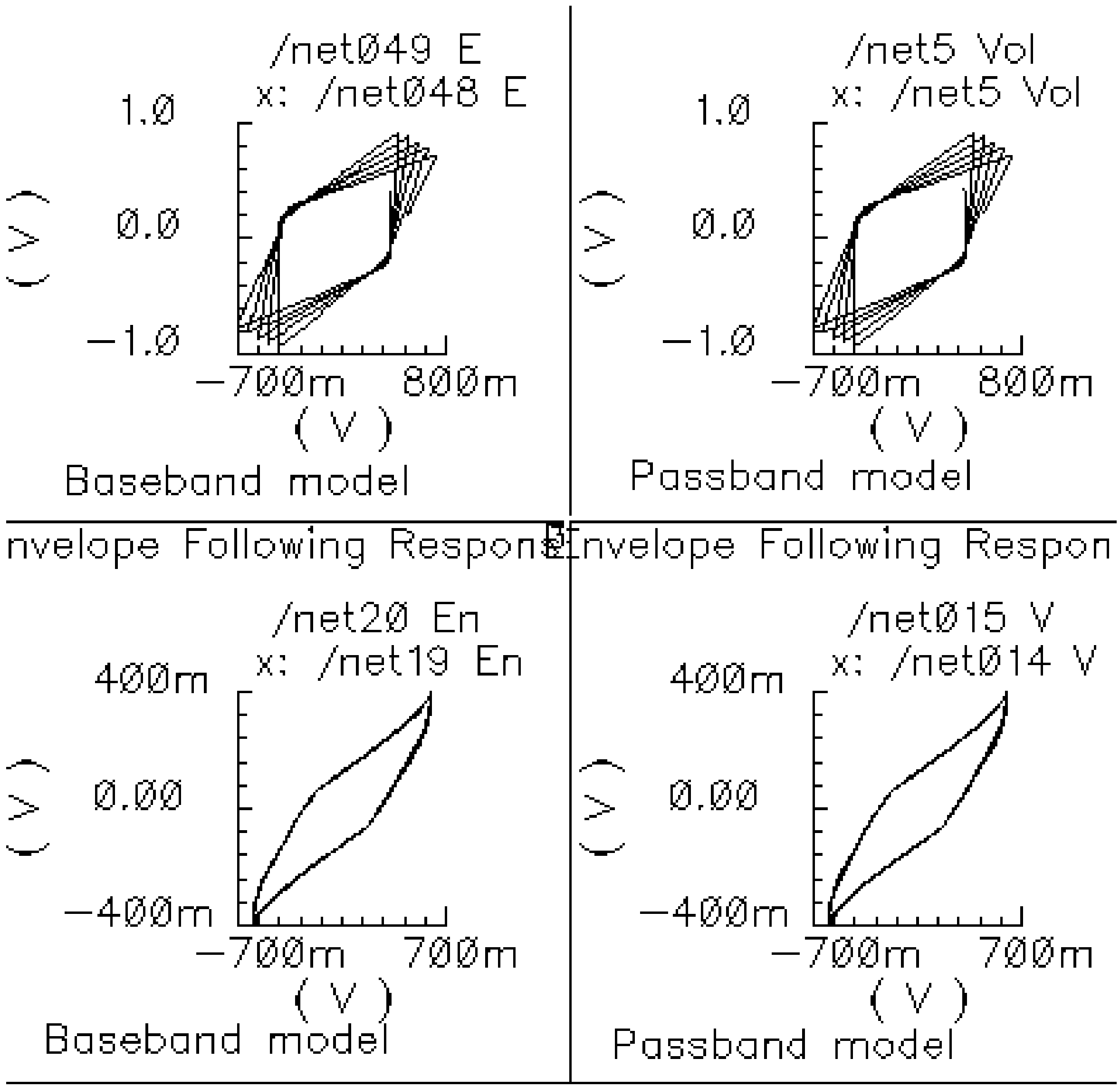
© 1994-2020 All Rights Reserved.

6. Open a fourth subwindow and plot the harmonic time results at the `IQ_demodulator` outputs but this time use the zero^{th} harmonic and only plot the real parts. Make the `I_out` waveform the x-axis. Figure 1-86 shows what you should now see.

The leftmost pictures are from the baseband models and the rightmost are from the passband models. Passband and baseband models agree quite well. The top pictures are the voltages at nodes that lie between the modulator and demodulator. Quadrature error squashes the baseband trajectory at that node. The trajectory precesses because phase error ramps up linearly with time just like in the last test. The non-linearities produce the sharp corners. The bottom trajectories do not precess because the same phase error rotates the demodulator output in the reverse direction; driven by the same phase error ramp, the demodulator undoes the precession introduced in the modulator. The demodulator outputs are nearly in phase because the quadrature errors of pi/4 in the modulators and demodulators add to give a total quadrature error of $\pi/2$, which in this case puts the baseband I and Q outputs nearly in phase with each other.

© 1994-2020
All Rights Reserved.

Figure 1-86 mod_demod Results



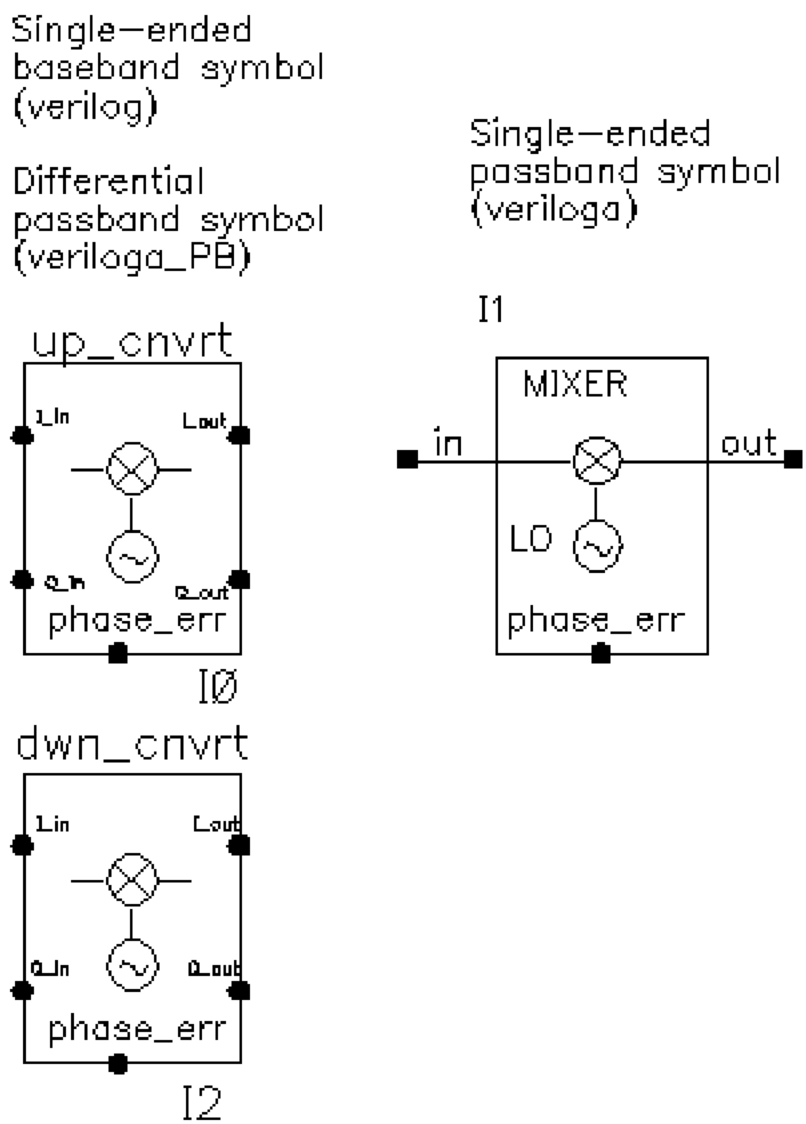
© 1994-2020
All Rights Reserved.

dwn_cnvt_test

(RF-to-IF and IF-to-RF Mixers)

(`passband = MIXER_PB, baseband = dwn_cnvrt` and `up_cnvrt`)

Figure 1-87 Baseband and Passband Mixer Models



`MIXER_PB` is a passband model that converts RF to IF and IF to RF. `dwn_cnvrt` model is a baseband equivalent model of a mixer used to convert from RF to IF. `up_cnvrt` model is a baseband equivalent model of a mixer used to convert from IF to RF. There are some minor

© 1994-2020
All Rights Reserved.

differences in the baseband models that depend on whether conversion is up or down. Figure 1-88 on page 184 and Figure 1-89 on page 184 show what the models do.

Figure 1-88 Calculations for up_cnvrt Mixer

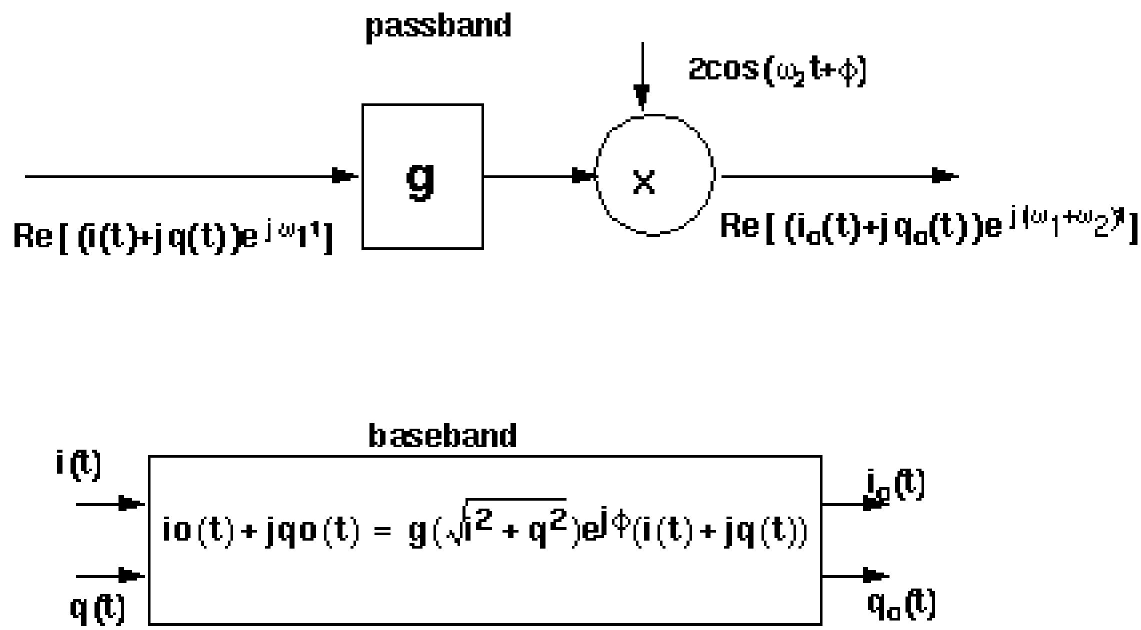
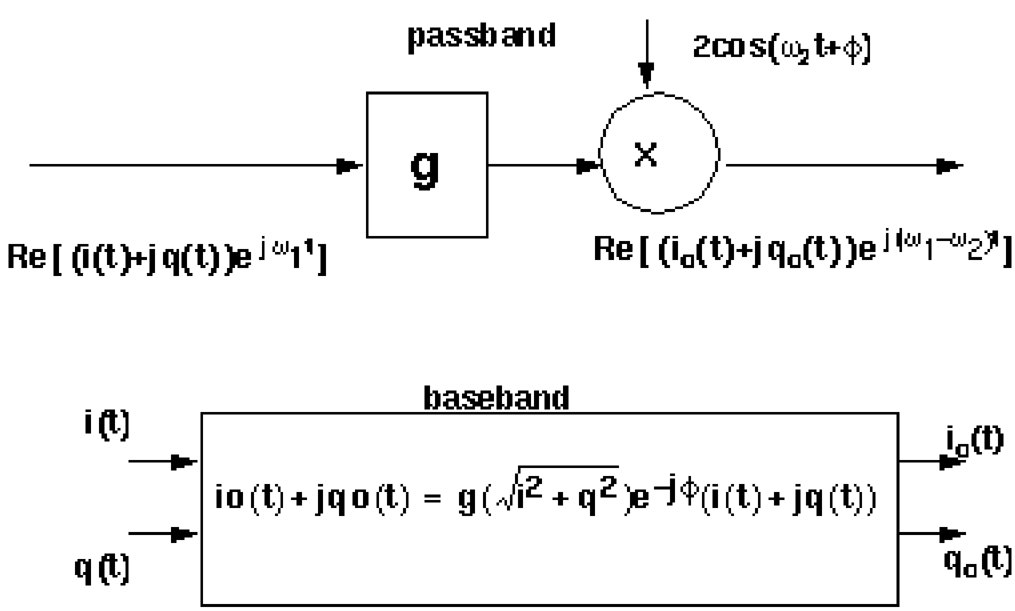


Figure 1-89 Calculations for dwn_cnvrt Mixer

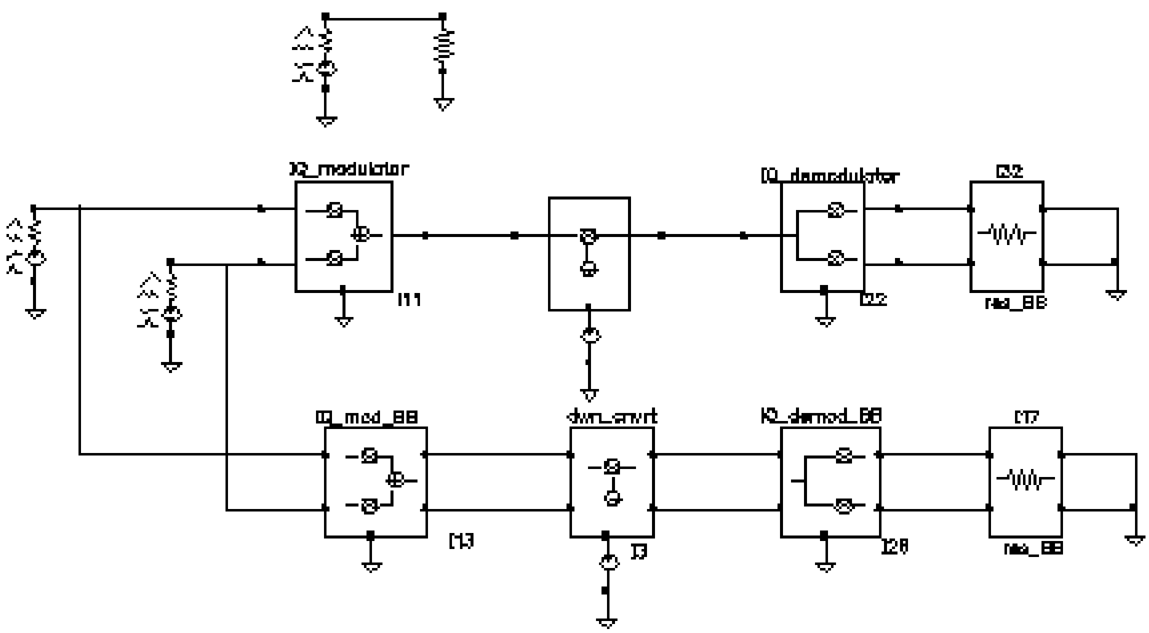


© 1994-2020 All Rights Reserved.

The noise figure and IP3 parameters are defined in "(IP3 Parameter)" on page 188. Unlike the `IQ_demodulator`, the IP3 test circuit can be used to define the available power gain because the gain is defined from the input frequency to just one sideband.

Typically the mixer would be used to create an IF stage. In that case, it is difficult to obtain a simple (i.e. filterless) envelope analysis that overlays waveforms to show how well baseband and passband models agree. The test circuit shown in Figure 1-90, which is listed as `dwn_cnvt_test` in the `testbenches` category of the `rfLib`, shows the relationship between baseband and passband models. The top branch of the circuit consists of passband models. The bottom branch consists of baseband models.

Figure 1-90 dwn_cnvrt_test Circuit

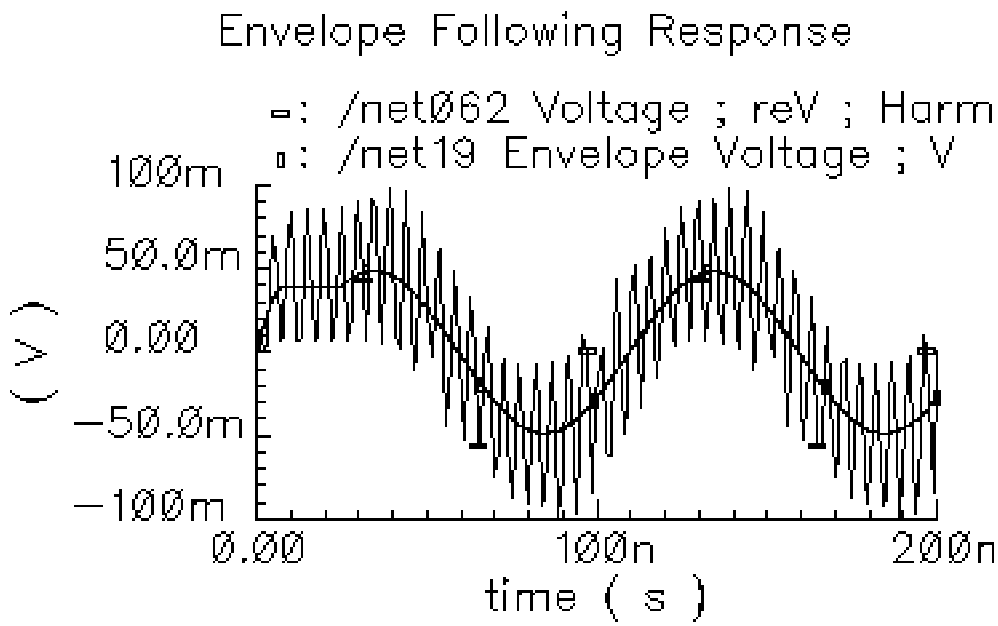


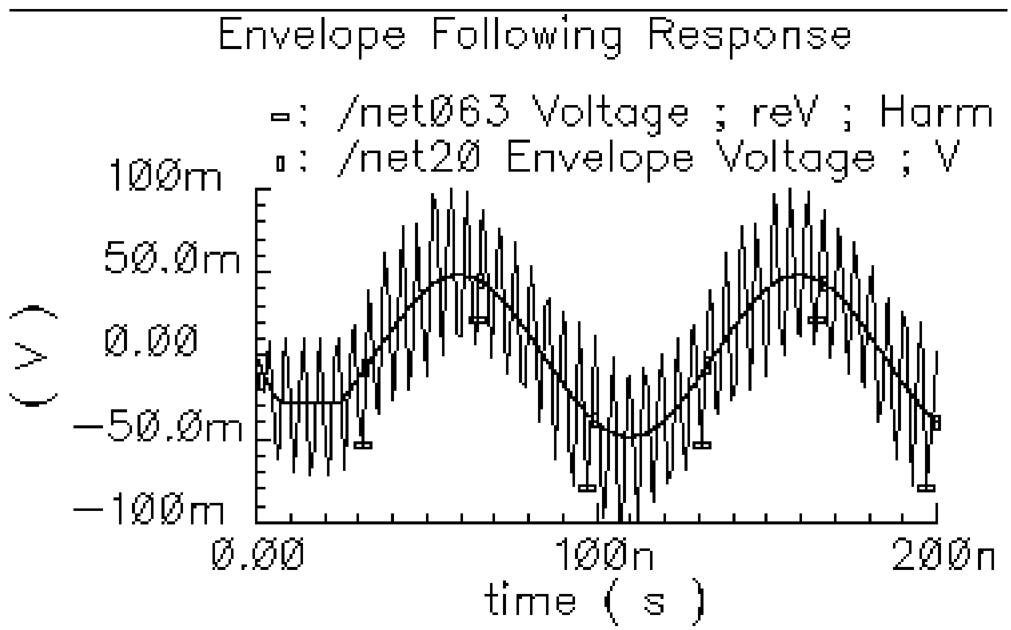
To see the relationship

1. Recall the circuit and set up a 200 ns envelope analysis with `fclck` as the *Clock Name*. Keep 1 harmonics1.
2. After the analysis completes, plot the "time" waveform at the `I_out` pin of the `IQ_demod_BB` model. Append to the plot, the harmonic-time, real part of the zero harmonic of the `I_out` pin on the `IQ_demodulator` model.
3. Open a subwindow and do the same for the `Q` outputs. You should now see a picture like the one in Figure 1-91.

© 1994-2020 All Rights Reserved.

Figure 1-91 Output from an Envelope analysis





To understand these results, trace the input signal through the passband branch. A complex baseband 10 MHz tone drives both branches. The modulator's local oscillator is 1 GHz so that the `IQ_modulator` output is at 1.01 GHz. There is no 990 MHz sideband because the input baseband trajectory is a circle (= sin + jcos), which represents a complex tone. The

© 1994-2020 All Rights Reserved.

mixer local oscillator is 900 MHz, which when mixed with 1.01 GHz, produces 110 MHz and 1.91 GHz. The `IQ_demodulator` local oscillator is 100 MHz, which produces 10 MHz, 210 MHz, 2.01 GHz, and 1.81 GHz. The 10 MHz and 210 MHz terms dominate the zero harmonic at the demodulator outputs. The higher frequencies average out to nearly zero. The baseband output is the 10 MHz term and that is what the baseband branch generates, as shown in Figure 1-91. A Transient analysis actually runs about 13 times faster than envelope on this circuit. Figure 1-116 compares the same outputs using a Transient analysis.The Transient analysis shows that the zero harmonic of the envelope analysis averaged out all frequencies above the envelope clock frequency (1 GHz).

© 1994-2020 All Rights Reserved.

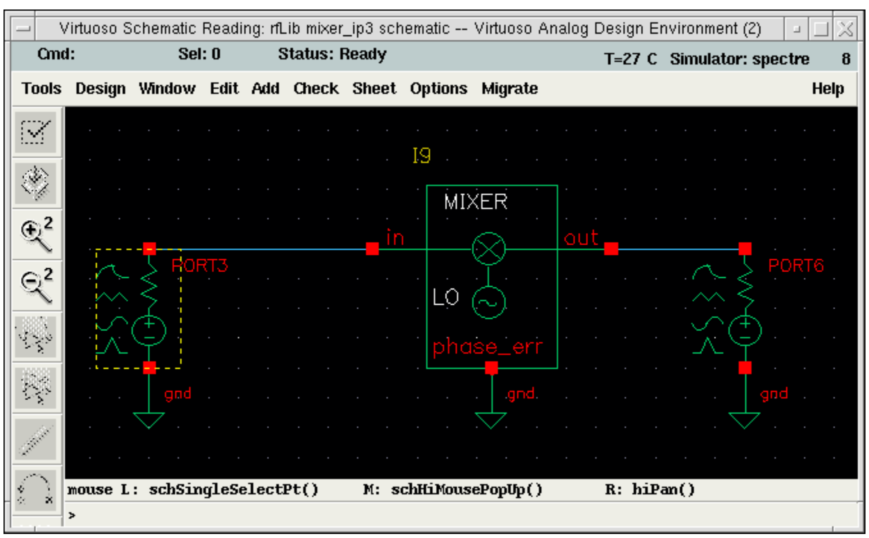
mixer_ip3

(IP3 Parameter)

IP3 is measured with a two-tone test. One tone is the fundamental PSS frequency while the other is the frequency in a single point PAC analysis. IP3 is defined as the input power level in dBm where the extrapolated power in one of the third order intermodulation terms equals the extrapolated power in the fundamental term. As with the 1dB compression point measurement, input and output terminals must be matched to the source and load respectively.

The IP3 specification is demonstrated step by step on the mixer model because the mixer IP3 measurement can be confusing. Figure 1-92 shows the test circuit. The circuit is listed as `mixer_ip3` in the `testbenches` category of the `rfLib`. For guidance on using the test circuit, see "Measuring IP3 for a Mixer" on page 188.

Figure 1-92 The mixer_ip3 Test Circuit



Measuring IP3 for a Mixer

For information about the test circuit used in this example, see "(IP3 Parameter)" on page 188.

1. Open the schematic for the circuit and bring up ADE.
2. In the Virtuoso Analog Design Environment window, choose *Analyses – Choose*.

   The Choosing Analyses form appears.

© 1994-2020 All Rights Reserved.

3. Set up a PSS analysis.

   a. Select *pss*.

      The title *Periodic Steady State Analysis* appears along with the fields required for specifying a PSS analysis.

      A 920 MHz tone already appears in the form.

   b. Add a *Fundamental Tone* called `eee` (the name is arbitrary) with a *Value* of 1 GHz.

   c. Select *Beat Frequency*.

   d. Click *Auto Calculate*.

      The result is `40M` Hz.

   e. For the *Number of harmonics*, type 2.

   f. Select *Sweep*.

   g. For the *Variable Name*, use the `power` variable.

   h. Set *Frequency Variable* to *no*.

   i. In the Sweep Range pane, select *Start_Stop*.

   j. In the *Start* field, type -60.

   k. In the *Stop* field, type 0.

   l. Select *Linear*.

   m. Select *Number of Steps*.

   n. In the *Number of Steps* field, type 10.

4. Set up a PAC analysis.

   a. Select *pac*.

      The title *Periodic AC Analysis* appears along with the fields required for specifying a PAC analysis.

   b. Set *Sweeptype* to *absolute*.

   c. Select *Single-Point*.

   d. In the *Freq* field, type 921 M.

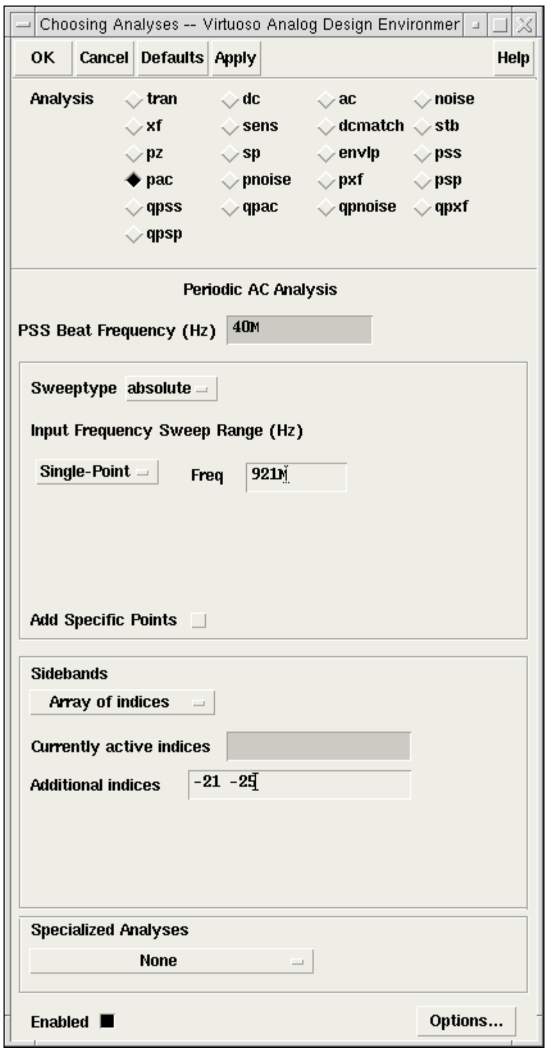
   e. In the *sidebands* pane, select *Array of indices*.

© 1994-2020 All Rights Reserved.

f. In the *Additional indices* field, type -21 and -25.

After these steps, the Choosing Analyses form looks like this.

Figure 1-93 Choosing Analyses PAC Setup

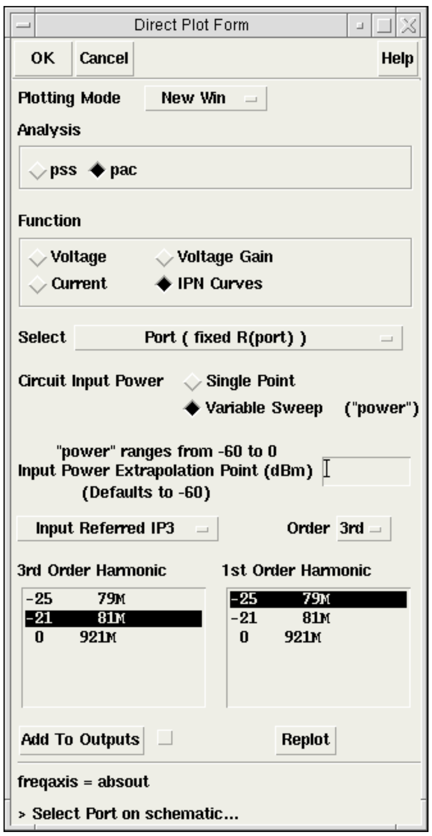


Why select the -21 and -25 sidebands? Recall from the assumptions, the non-linearity occurs before the frequency translation. The input tones to the non-linearity are the large 920 MHz tone and the small signal 921 MHz tone. In an IP3 measurement only one tone must be large, the other can be small. PAC analysis performs small signal perturbations on the PSS solution. One perturbation term exiting the non-linearity appears at 921 MHz, right where it started. One of the third order intermodulation perturbation terms exiting the non-linearity appears at 2*920-921 = 919 MHz. The ideal mixer, driven by a pure 1 GHz local oscillator, translates the 921 MHz tone to 921-

© 1994-2020
All Rights Reserved.

1000=-79 MHz while translating the 919 MHz tone to 1000-919=81 MHz. A single point 921 MHz PAC analysis produces tones displaced from harmonics of the fundamental by 921 MHz. The PAC sidebands specify which harmonics to use. You save the 79 MHz tone by saving the -25th sideband because the fundamental frequency is 40 MHz and 921 - 40*25 = -79 MHz. You save the 81 MHz tone by saving the -21 sideband because 921-40*21 = 81 MHz. Figure 1-93 on page 190 shows the PAC setup.

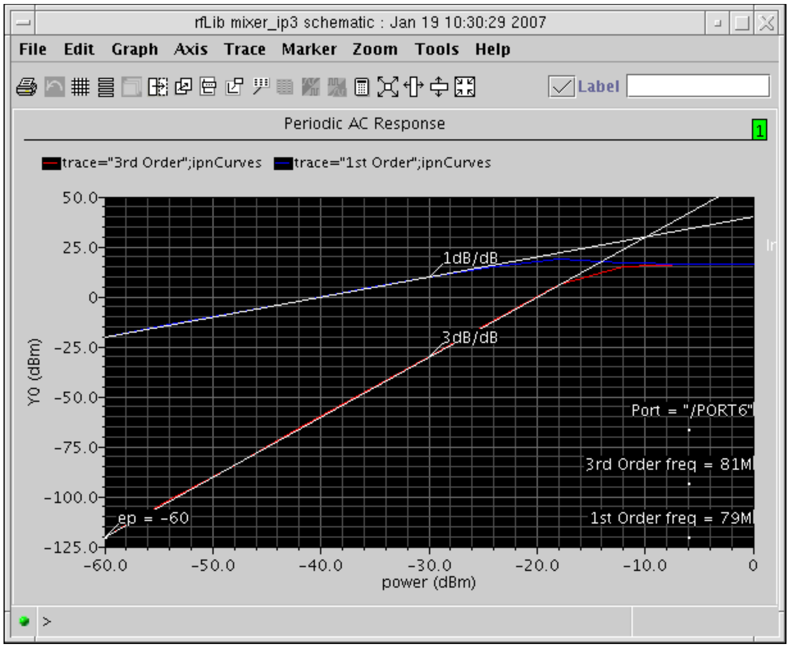
5. Run the analysis.
6. Plot the PAC results. To do this, set up the Direct Plot form like this.



7. In the Composer window, click the output Port. The results appear as shown in Figure 1-94 on page 192.

© 1994-2020
All Rights Reserved.

Figure 1-94 IP3 Results



The measured IP3 is, -10 dBm, as specified. The measured IP3 is as specified only if the input port resistance matches the input resistance of the device-under-test. Other input resistances produce a measured IP3 different than the one specified.

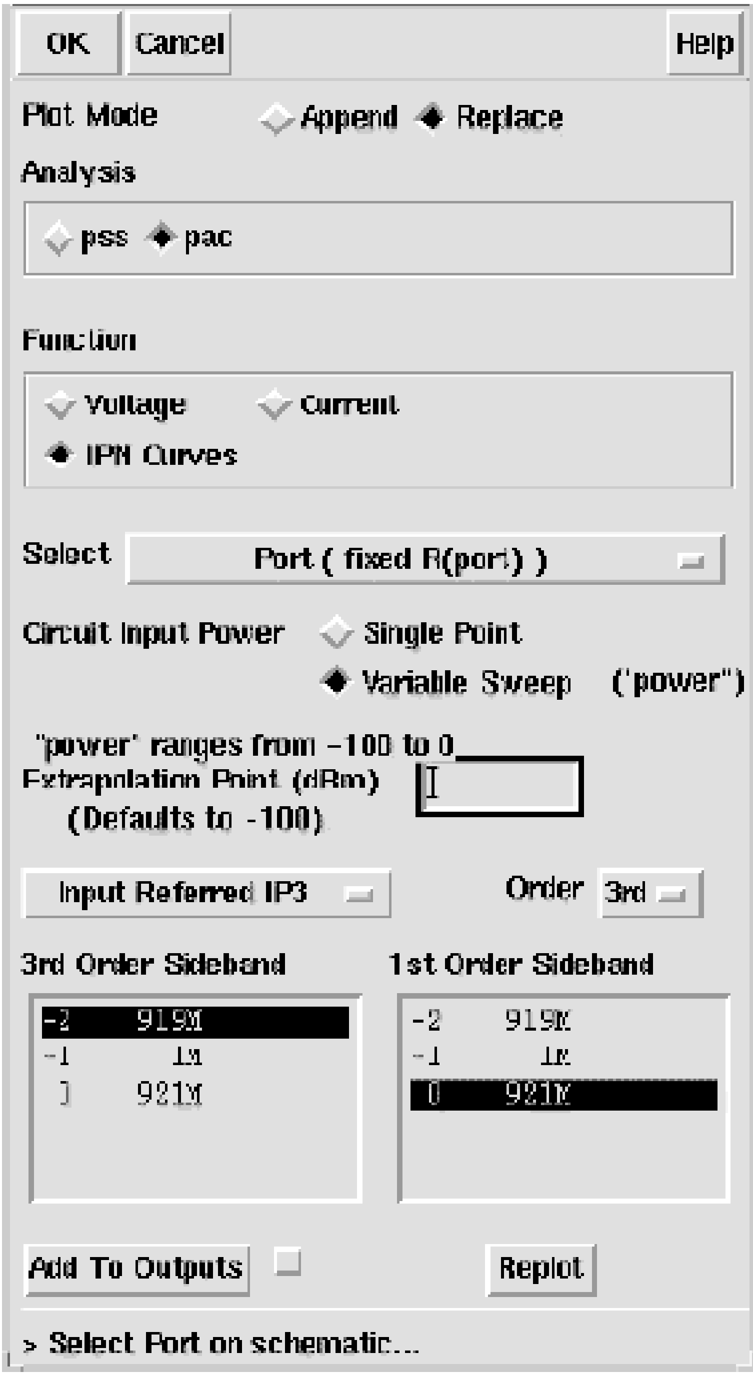
Measuring IP3 for an LNA

You can measure IP3 of an LNA by replacing the mixer with an LNA and ensuring the input terminal remains matched. In this example, remove the 1 GHz *Fundamental Tone* from the PSS analysis. The *Beat Frequency* should now be 920 MHz. In the PAC set up, change the additional indices from -21 and -25 to -1 and -2.

After the analysis completes, set up the PSS Results form as shown in Figure 1-95. As in “Measuring IP3 for a Mixer” on page 188, the input referred IP3 is 10 dBm, as specified. Figure 1-96 shows the LNA IP3 results.

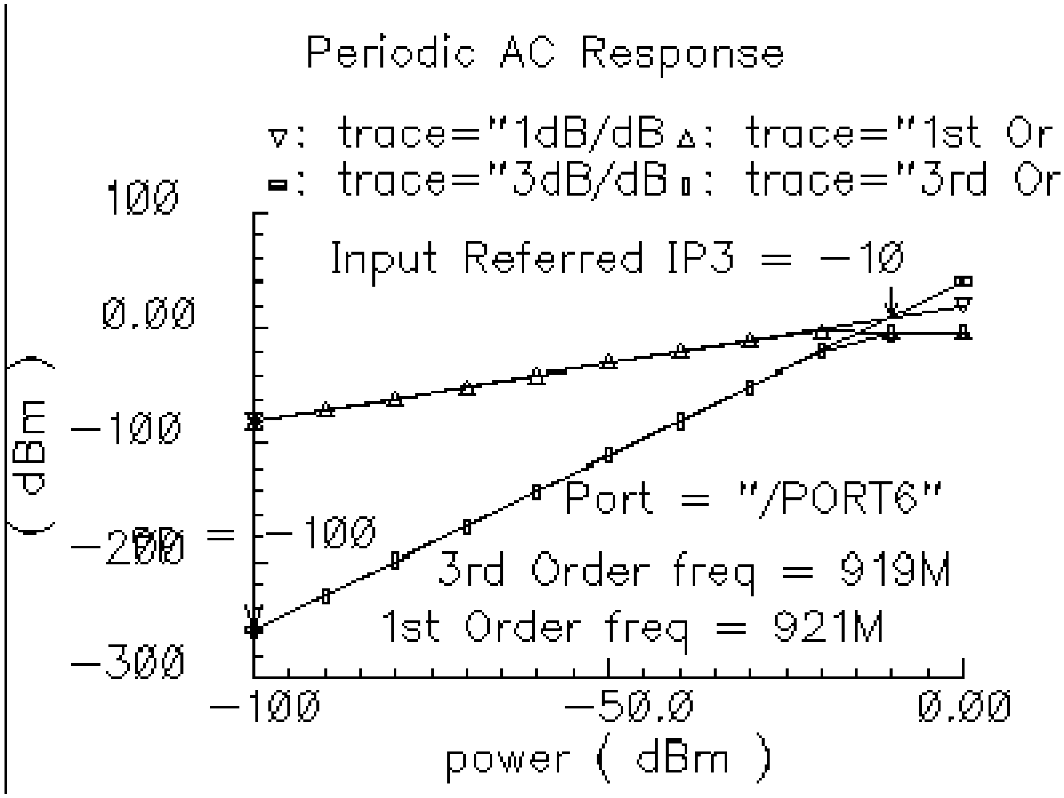
© 1994-2020 All Rights Reserved.

Figure 1-95 Direct Plot Form for the LNA



© 1994-2020 All Rights Reserved.

Figure 1-96 Results for the LNA



All Rights Reserved.

mod_1dbcp

(Available Power Gain and 1dB Compression Point)

Available power gain of the `IQ-modulator` is best explained with an example. Recall the circuit called `mod_1dbcp` listed in the `testbenches` category of the `rfLib`. The schematic contains two disjoint circuits. One shows how not to measure gain and compression point, the other shows the proper measurement.

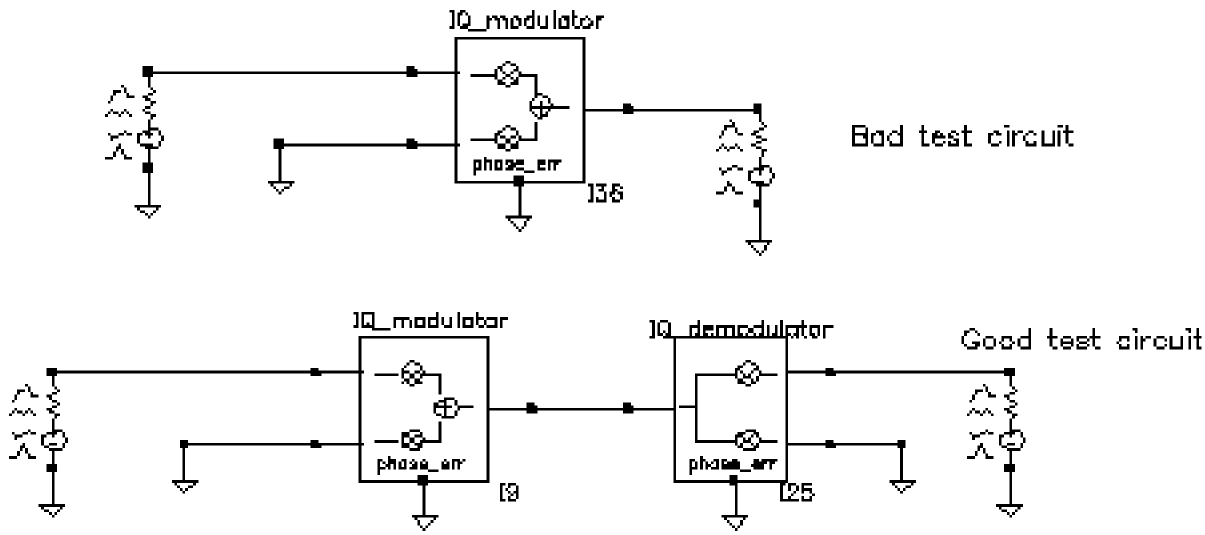
1. Set up a PSS analysis. Both test circuits run in the same simulation. The beat frequency is 100 MHz. Save the first and 11th harmonics. In the options, set `maxstep` to 50 ps. Sweep the variable `power` linearly in 50 steps from -40 to 15.

2. After the analysis completes, plot the output referred 1dB compression point of the top circuit using -40 dBm as the Extrapolation point. First select the 11th harmonic (1.1 GHz) and click the output port in the top test circuit, the bad test circuit. Note that the linear gain is 3 dB lower than specified, as is the output referred 1dB compression point. The gain was specified as zero dB and the 1dB compression point was 10 dBm. The error arises from the fact that the input signal power splits between upper (1.1 GHz) and lower (900 MHz) sidebands but the ADE measurement only looks at one output sideband. The bottom test circuit resolves the ambiguity by defining the gain of the IQ-modulator as the gain from the baseband input to an ideally-demodulated baseband output. The bottom test circuit follows the IQ-modulator with an ideal IQ-demodulator. The gain of the demodulator is zero dB and the 1dB compression point is high enough to render the demodulator distortionless.

3. Repeat the steps for plotting the 1dB compression point but this time chose the first harmonic and select the output port that loads the bottom circuit. Select the first (100 MHz) harmonic and plot the 1dB compression point again. Now you should see a 1dB compression point plot that reflects the specified parameters of the IQ-modulator. The gain is now also correct, which can be computed from the ratio of the output to input power well below the compression point. Figure 1-98 shows such a plot.

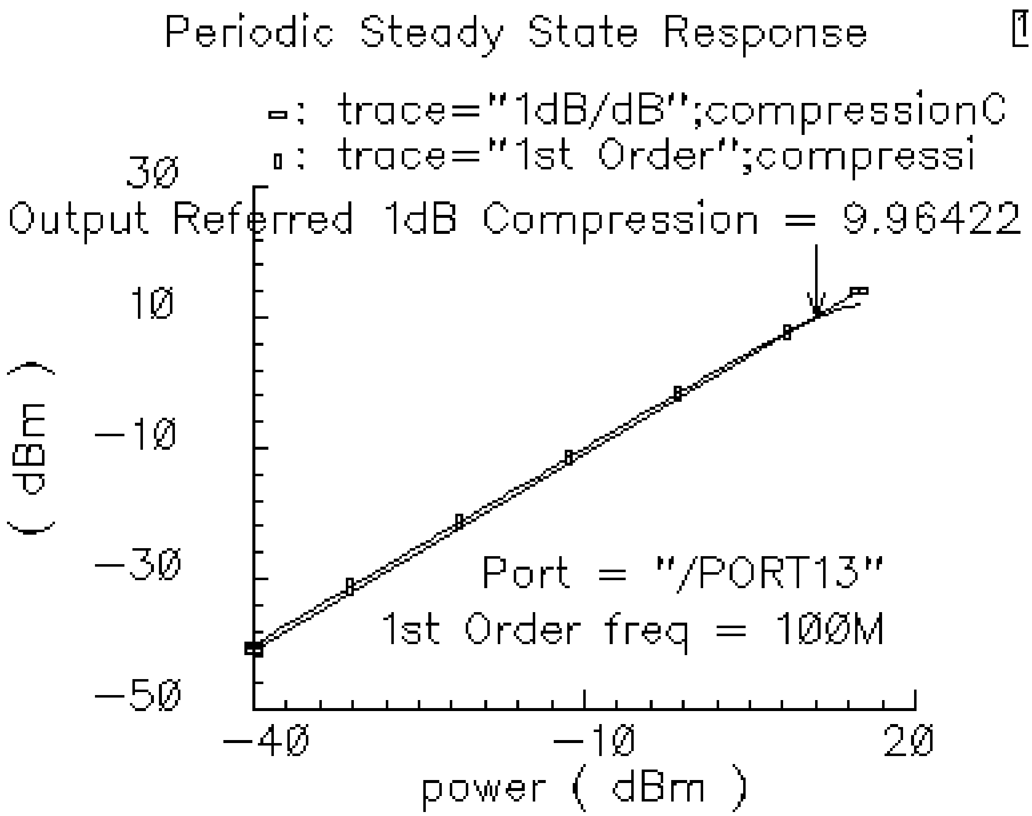
© 1994-2020 All Rights Reserved.

Figure 1-97 1db Compression Point Test Circuit



© 1994-2020 All Rights Reserved.

Figure 1-98 1db Compression Point Plot



All Rights Reserved.

mod_demod_test

See "demod_ip3" on page 175.

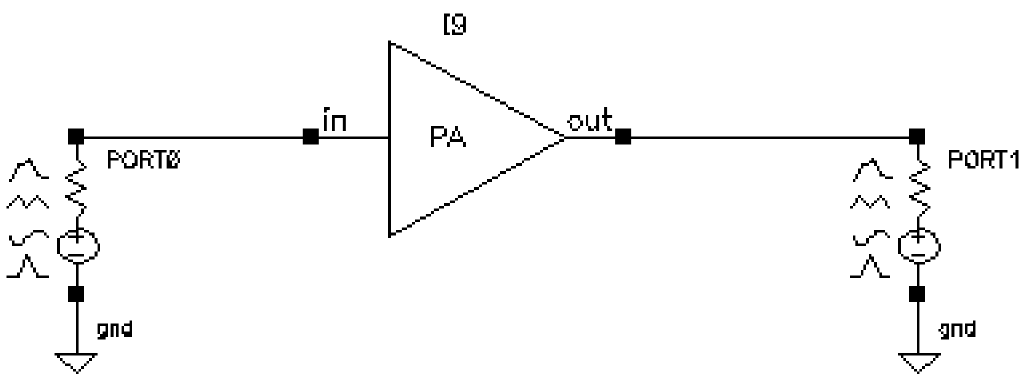
© 1994-2020 All Rights Reserved.

noise_figure

(Noise Figure Parameter)

Noise figure is calculated as the input signal-to-noise ratio divided by the output signal-to-noise ratio. The test circuit for defining the noise figure parameter is shown in Figure 1-99. The circuit is listed as `noise_figure` in the `testbenches` category of the `rfLib`. It is similar to the `one_db_cp` test circuit.

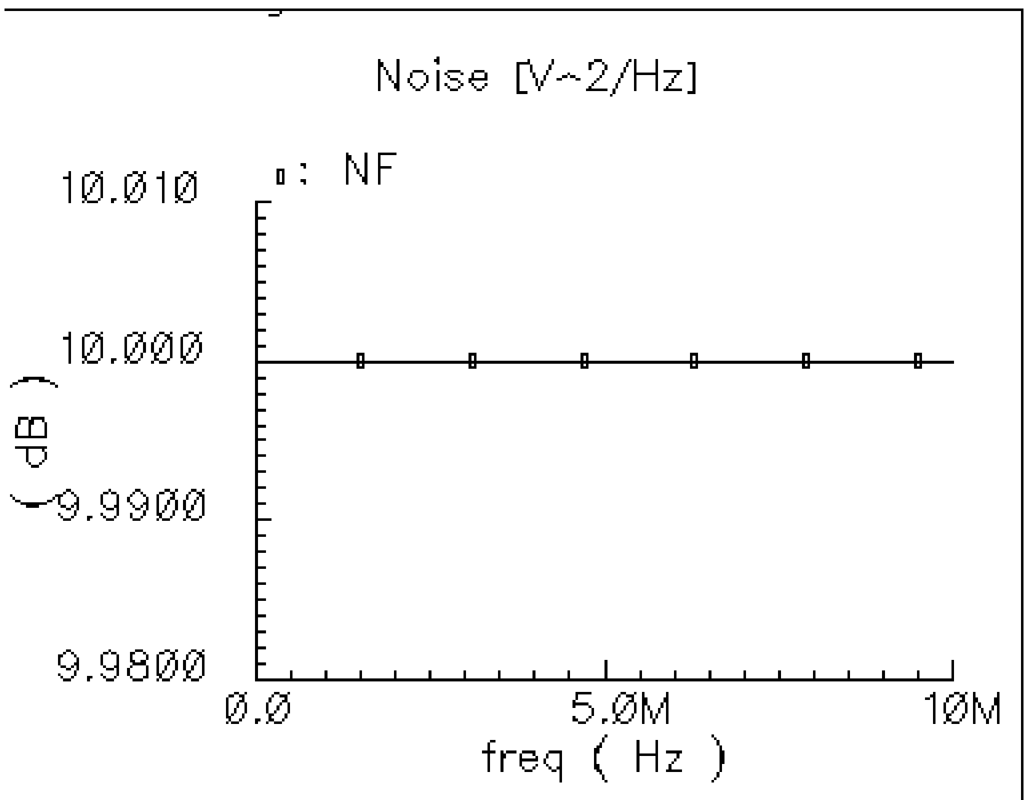
Figure 1-99 The noise_figure Circuit



The specified noise figure is 10 dB. A Spectre RF Noise analysis produces the noise figure shown in Figure 1-100. To measure the specified noise figure, the driving port resistance must match the amplifier's input resistance. The port at the output does not have to match the amplifier's output resistance but the port impedance should be resistive. The input probe is the leftmost port, the output port is the rightmost port. Because the model is static, you can compute noise figure over any frequency interval.

© 1994-2020 All Rights Reserved.

Figure 1-100 Noise Figure Results



All Rights Reserved.

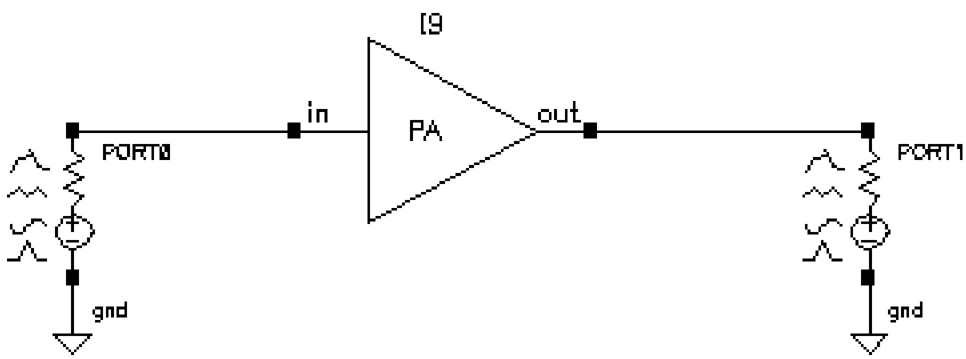
one_db_cp

(Output 1dB Compression Point Parameter)

The 1 dB compression point specifies a saturation non-linearity. It is the output power in dBm where the output power falls 1 dB below the power extrapolated linearly from the amplifier’s linear region of operation.

The test circuit in Figure 1-101 is listed as `one_db_cp` in the `testbenches` category in `rfLib`.

Figure 1-101 The one_db_cp Circuit

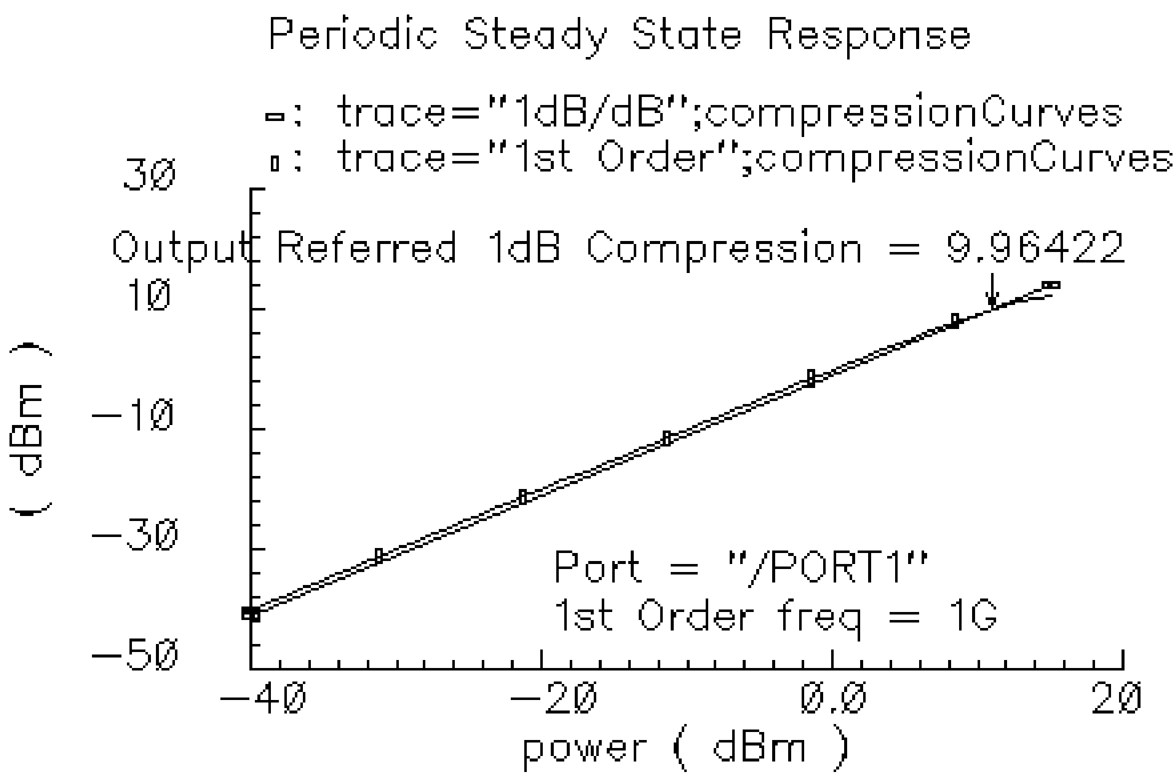


In the `one_db_cp` test circuit, *power* is the dBm of power delivered by the leftmost port. The available power gain is 0 dB. The 1dB compression point is 10 dBm. The input and output resistances are 50 Ohms and so are the port resistances.

To measure the 1dB compression point, perform a swept PSS analysis. Sweep *power* from -40 dBm to 15 dBm in 50 linear steps. The output referred 1dB compression point is computed for the 1st harmonic with an *Extrapolation Point [dBm]* of -40. Click the rightmost port device to display the output as illustrated in Figure 1-102.

© 1994-2020 All Rights Reserved.

Figure 1-102 Resulting 1dB Compression Point



The specified output referred compression point is 10 dBm. The measured value is 9.964 dBm, which is fairly close to the specified value. The measured 1db compression point is as specified only when the driving source resistance matches the amplifier input resistance and the load port resistance matches the amplifier's output resistance. In all compression point and IPN calculations, input power is computed from the maximum power the input Port can deliver, not from an actual power measurement. If you mismatch either terminal you do not measure the specified compression point.

© 1994-2020 All Rights Reserved.

PB_BB_filter_comparison

See “BB_ind_cap_test” on page 169.

© 1994-2020 All Rights Reserved.

PB_ind_cap_test

See "BB_ind_cap_test" on page 169.

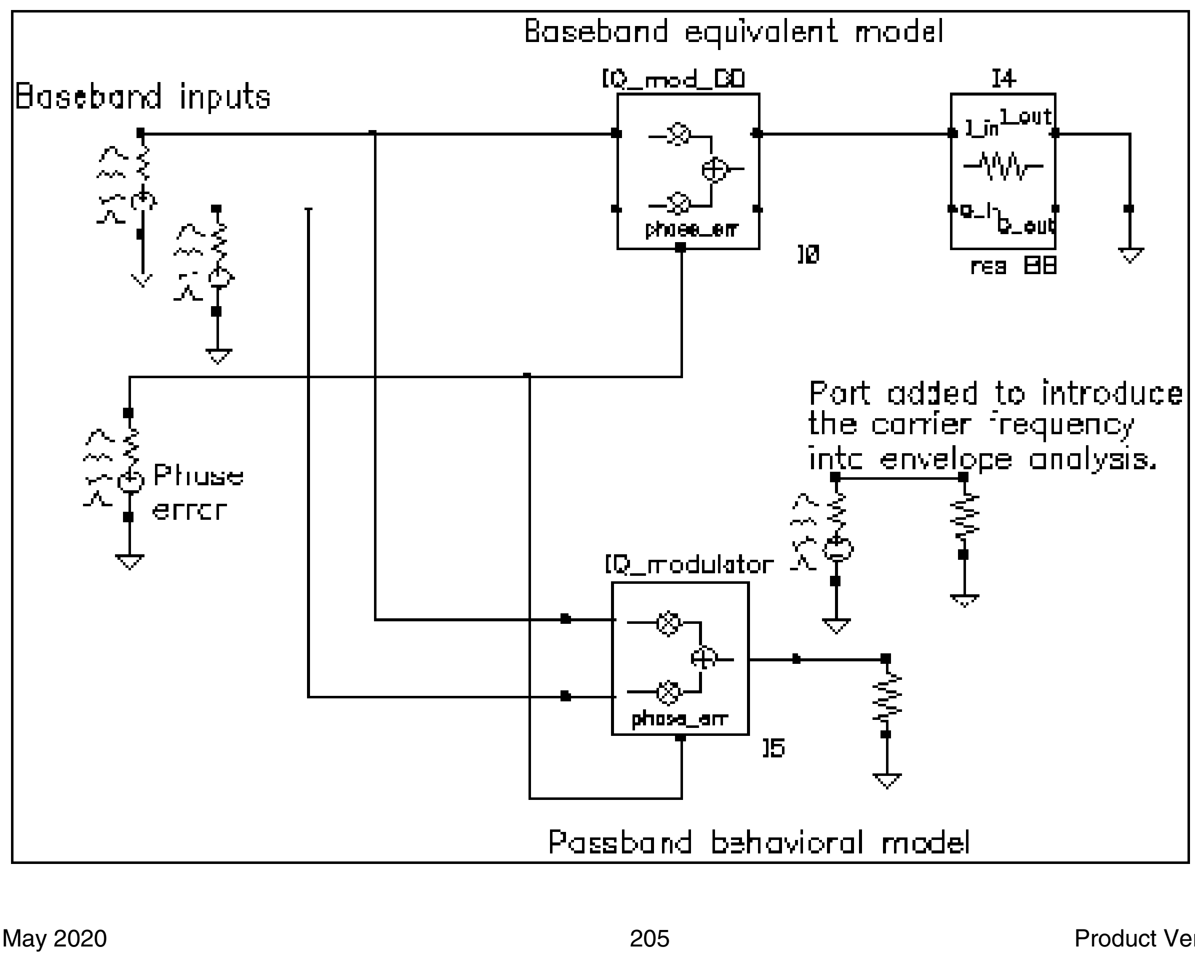
© 1994-2020
All Rights Reserved.

quad_and_phase_error_demo

(Quadrature Error and Phase Error)

Quadrature error describes how far away from 90 degrees the two local oscillators are from each other. Ideally, they are exactly 90 degrees, or π/2 radians, apart in phase. In practice, parasitics and asymmetric delays can drive the phase shift away from π/2. Figure 1-103 on page 205 show a baseband test circuit and its passband equivalent. The schematic is listed in the `rfLib testbenches` category under the name `quad_and_phase_error_demo`. Both circuits are driven from a common set of baseband sources. The test circuit serves two purposes, it shows the correspondence between baseband and passband models and it demonstrates how quadrature error and phase error affect the baseband trajectory. The baseband input signal is a complex tone, which makes a circular input baseband trajectory. If there were no quadrature error, the baseband representation of the modulator output would also be a circle. With quadrature error, the output trajectory is an ellipse. If the `phase_err` pins are driven by a ramp, the ellipse precesses. The ramp represents a small but fixed difference between carrier and local oscillator frequencies.

Figure 1-103 quad_and_phase_error_demo Circuit

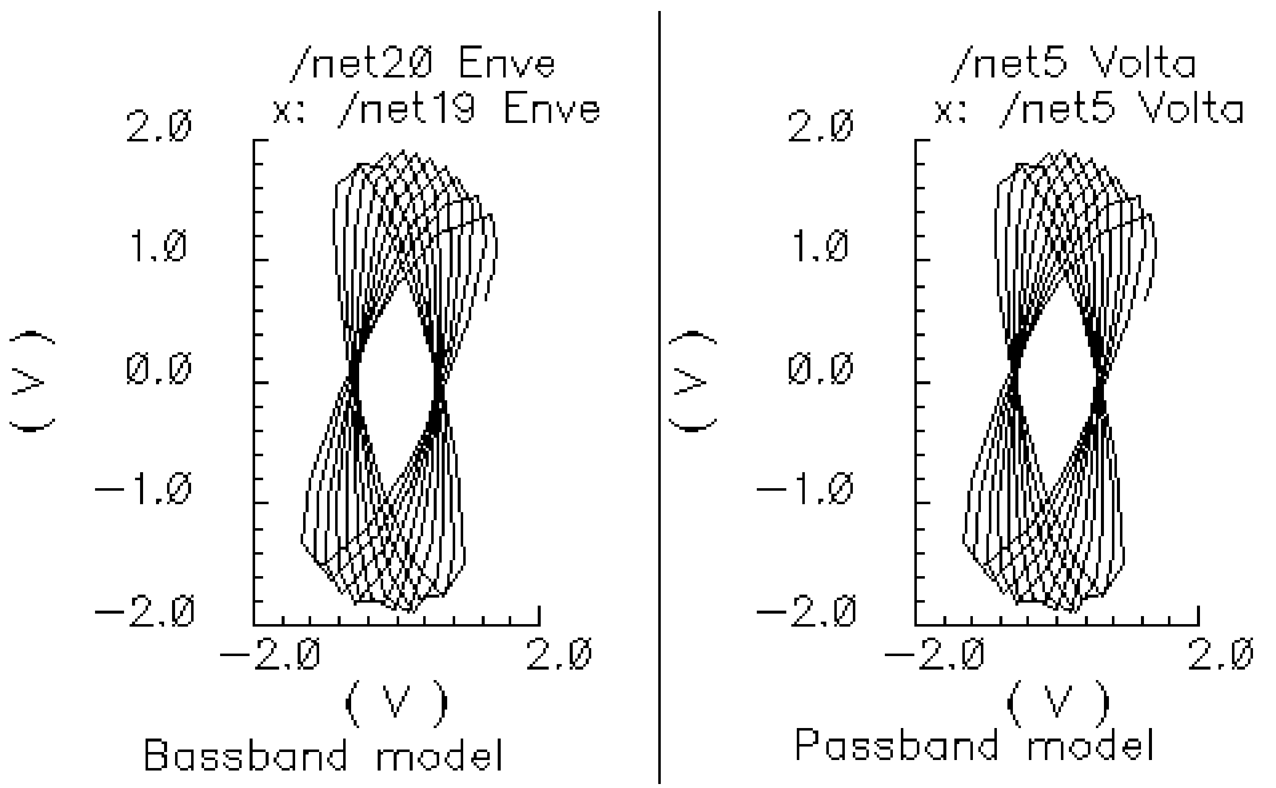


© 1994-2020 All Rights Reserved.

To see these effects

1. Set up an ENVLP analysis with `carrier` as the Clock Name. Simulate 10 us of action and save the first harmonic.
2. When the analysis completes, open the Envelope Following Direct Plot form and set the sweep to *time*. Plot the two outputs of the `IQ_mod_BB` block.
3. Change the x-axis to be the `I-output`. You should see the left trajectory in Figure 1-104.
4. Add a subwindow for the passband equivalent result.
5. In the Direct Plot form, change the sweep to *harmonic time* and plot the real and imaginary parts of the first harmonic of the `IQ_modulator` output voltage.
6. Change the x-axis to be the real part of the first harmonic. You should now have two plots that match those in Figure 1-104.

Figure 1-104 Output Trajectories



All Rights Reserved.

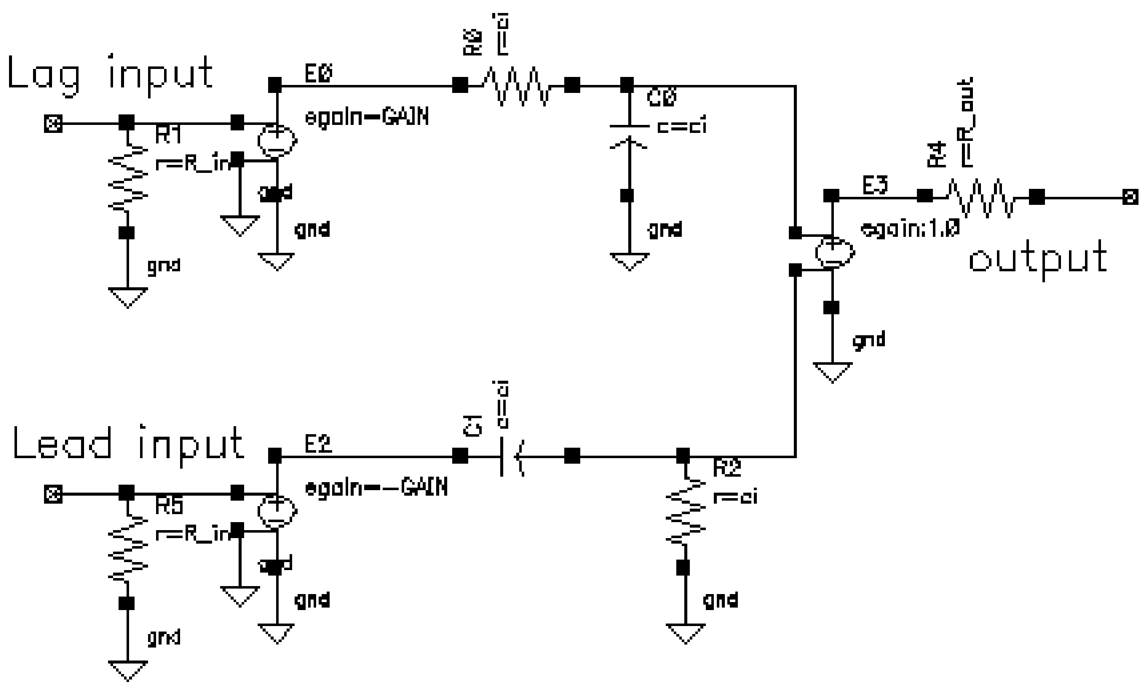
shifter_combiner_test

(Phase Shifter Combiner)

`(shifter-combiner)`

The phase shifter-combiner has two inputs and one output. The inputs are phase shifted by +/- 45 degrees then added together to form the output. All terminals are buffered and have the specified terminal resistances. The phase shifts are accomplished with Verilog-A code that does the same thing as the circuit shown in Figure 1-105 on page 207. The gains of the left-most voltage-controlled-voltage sources are user-defined. The input resistance, output resistance, intended operating frequency, and internal resistance are also user-defined. The internal resistance and operating frequency are used to calculate the capacitance necessary to provide 45 degrees of phase shifts at the operating frequency. The baseband view requires that the carrier frequency be specified.

Figure 1-105 Phase Shift Circuit

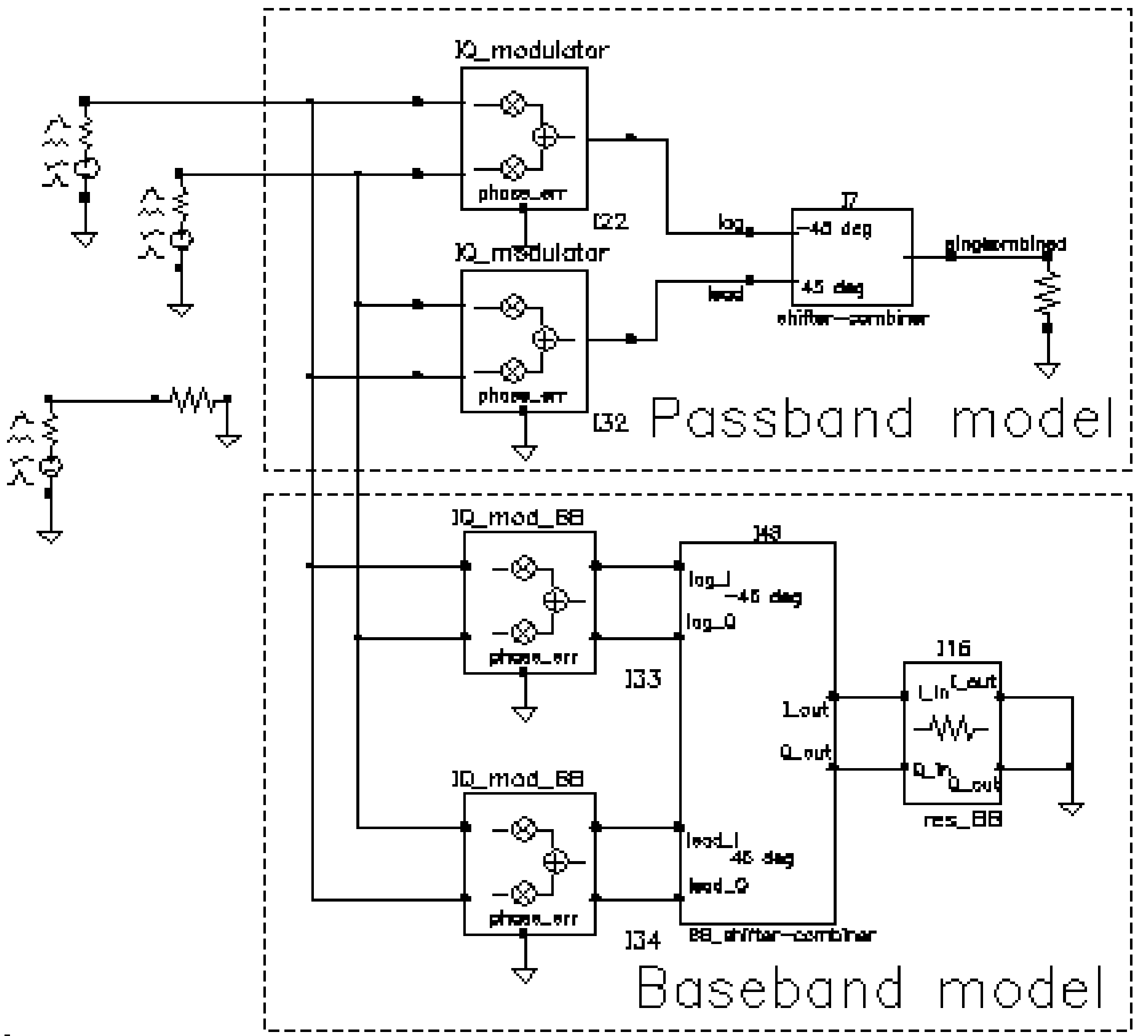


The shifter-combiner can be used to eliminate one phase of the carrier. The test circuit in Figure 1-106 on page 208 shows a simple test to demonstrate the idea. The circuit is in the `rfLib` under the `testbenches` category and listed as `shifter_combiner_test`. The top circuit is a passband model and the bottom circuit is the baseband equivalent. Baseband input signals are mixed up to 1GHz then passed into the shifter-combiner. The baseband

All Rights Reserved.

signal contains 10MHz and 20Mhz components. The modulators and shifter-combiner are arranged to produce only a 20MHz signal riding on the carrier.

Figure 1-106 shifter_combiner_test Circuit



To check this assertion:

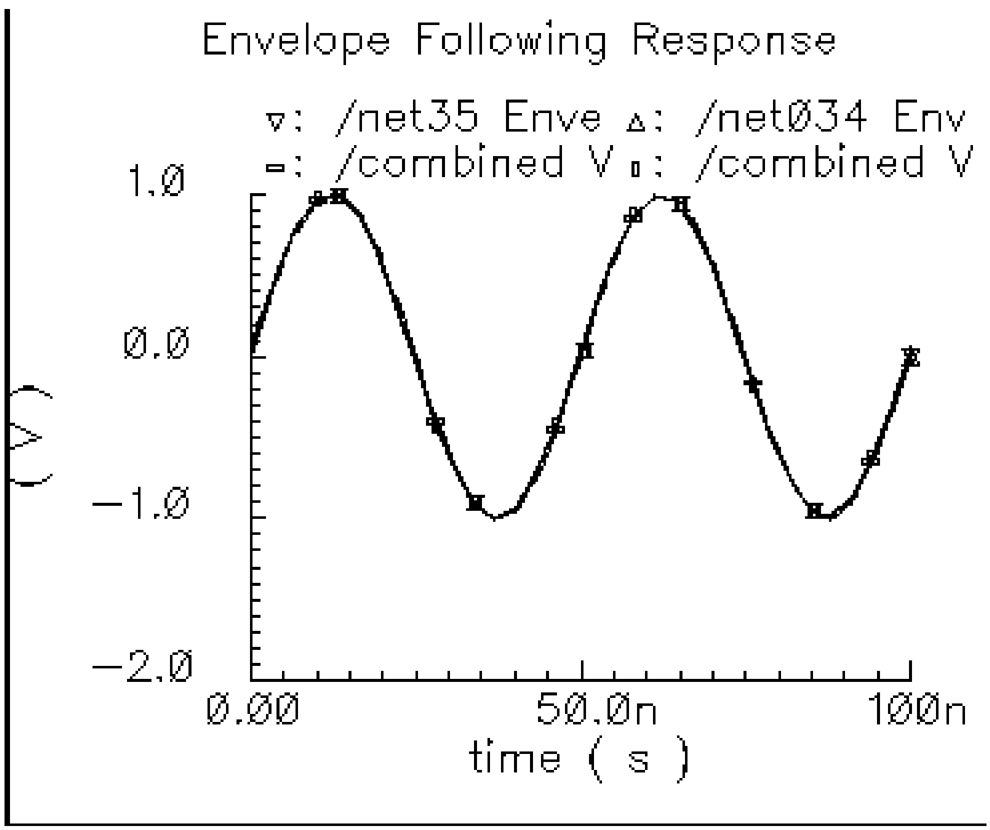
1. Bring up the test circuit and an Analog Environment window.
2. Set up a 100ns envelope analysis on the circuit with the *Clock Name* set to `carrier` and the *modulationbw* option set to 40MHz. Set the *Harmonic number* to 1.
3. Run the analysis.

© 1994-2020 All Rights Reserved.

4. Plot the harmonic time, 1 Harmonic, and the real and imaginary parts of the passband shifter combiner output.

5. Append to the plot, the `time` waveforms at the `I_out` and `Q_out` pins of the `BB_shifter _combiner` model. Figure 1-107 on page 209 shows what you should see in the Waveform Display window. All waveforms are the same and they contain only the 20Mhz baseband signal. The 10Mhz baseband input signal does not propagate to the output.

Figure 1-107 shifter_combiner_test results



One application of the shifter-combiner is an image rejection receiver. Figure 1-108 on page 210 shows a very simple example of an image rejection receiver. Figure 1-109 on page 210 shows the baseband equivalent model of the receiver. Both examples are in the `rfExamples` directory and are listed as `image_reject_rcvr_PB` and `image_reject_rcvr_BB`. The local oscillator runs at 1GHz and the RF carrier is 1.1GHz, which places the image at 900Mhz. This example shows one of the limitations of the baseband equivalent models.

© 1994-2020 All Rights Reserved.

Figure 1-108 A Simple Image Rejection Receiver

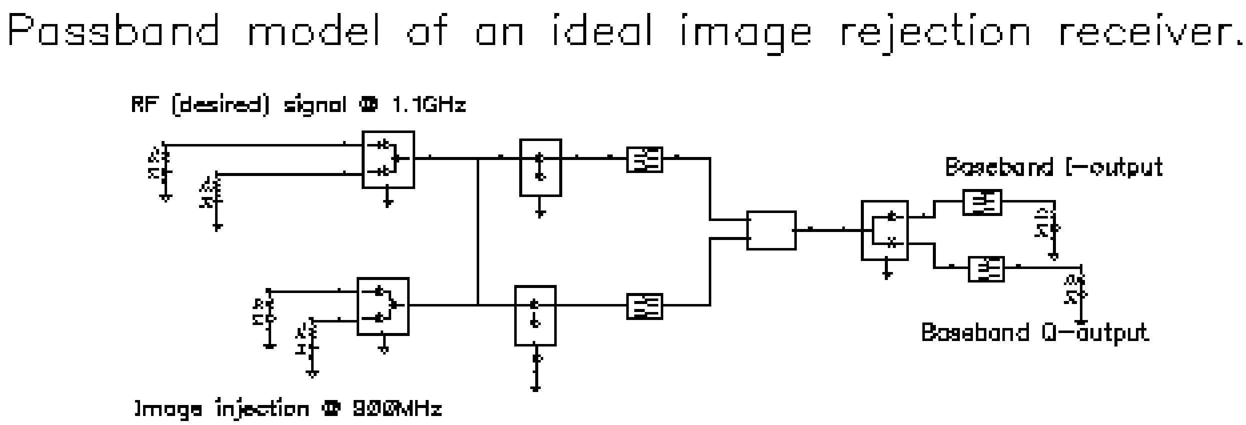
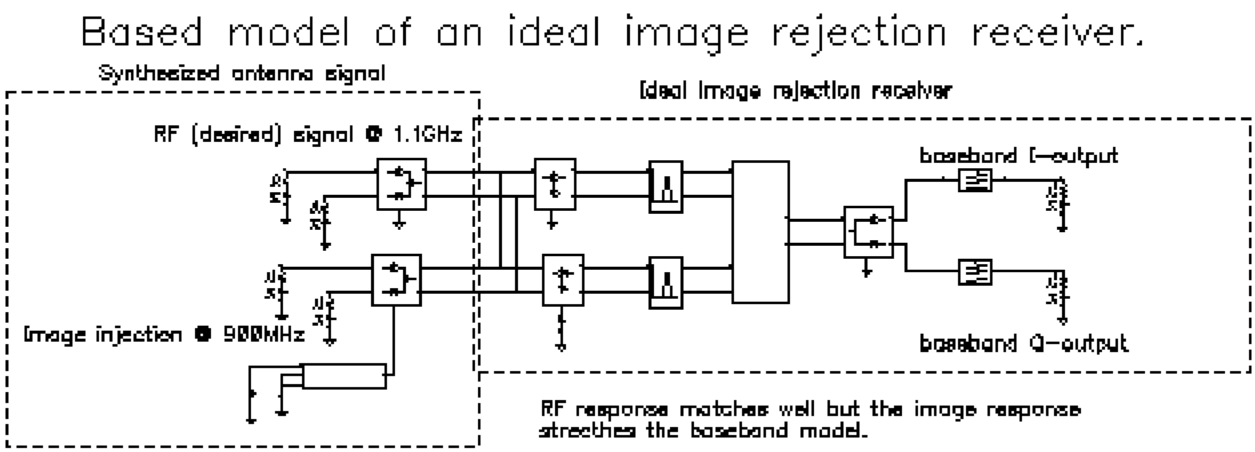


Figure 1-109 Baseband Equivalent Model of the Image Rejection Receiver



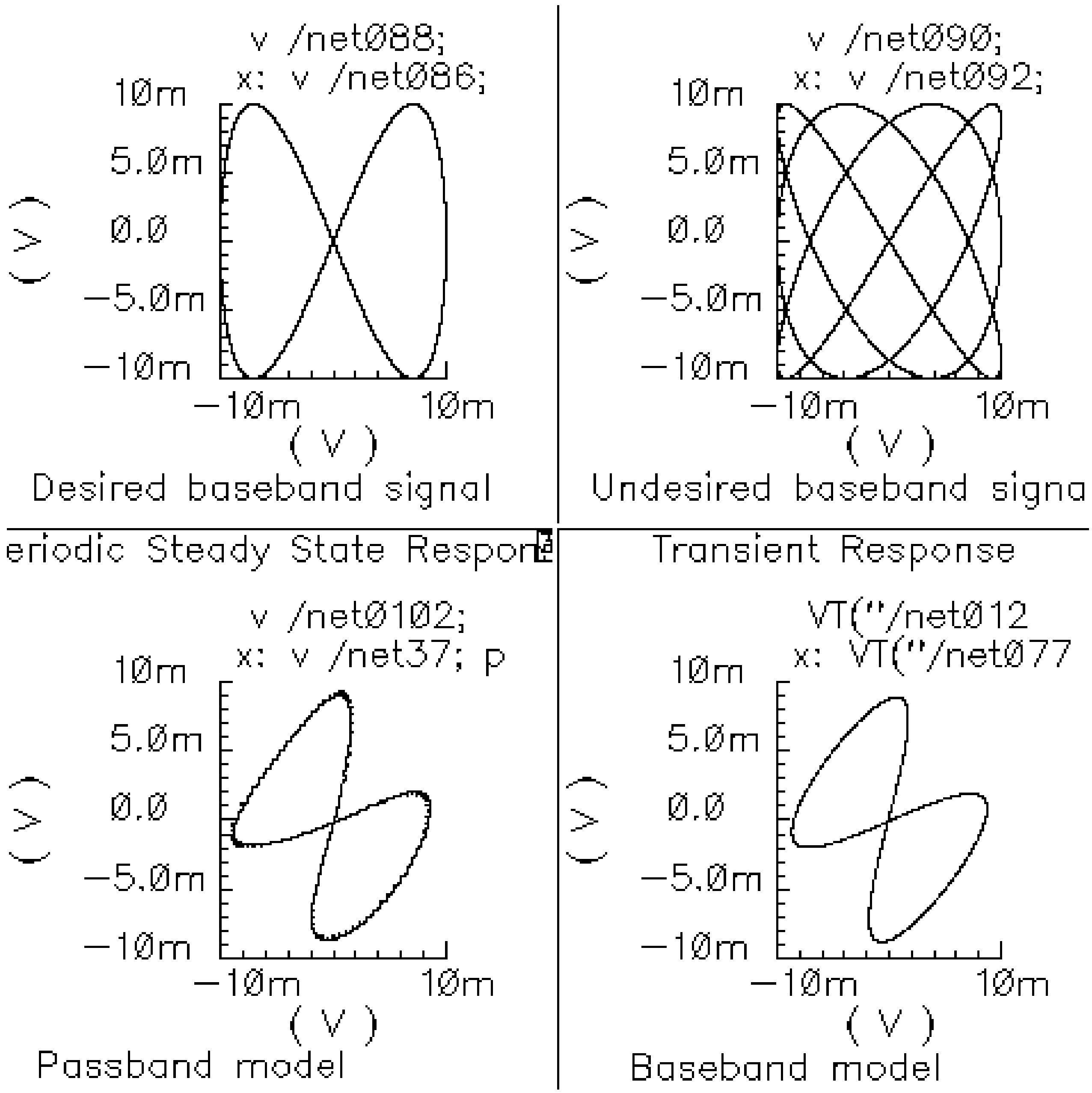
1. Bring up the passband test circuit and an Analog Environment tool.
2. Set up a PSS analysis. You need to add the 1.1GHz, 1GHz, and 900Mhz fundamental tones. Give them arbitrary but distinct names. AutoCalculate the Beat Frequency, which should be 2MHz. You need not save more than the 1st harmonic. Set the PSS maxstep option to 20ps so that it accurately simulates the oscillators hidden inside the Verilog-A modules.
3. Run the analysis.
4. Plot the voltages across Ports 5 and 6. Set the x-axis to be the voltage across Port 6. This is a Lissajous plot of the desired baseband signal, the one riding on the 1.1GHz carrier.

© 1994-2020 All Rights Reserved.

5. Add a subwindow.
6. Plot the voltages across Ports 8 and 7. Set the x-axis to be the voltage across Port 8. This is a Lissajous plot of the undesired baseband signal, the signal riding on the image of the carrier at 900MHz.
7. Add another subwindow.
8. Plot the I and Q- baseband outputs. Set the x-axis to be the I-output. The Lissajous plot is a tilted version of the desired baseband signal, indicating that most of the image was successfully rejected.
9. Bring up the baseband equivalent receiver model and another Analog Environment tool.
10. Run a 10us Transient analysis with 9.5us as the output start in the analysis options and maxstep set to 250ps. The phase_err pin on the image signal generator is being driven to spin the output at 200MHz, the frequency difference between the desired frequency and image frequency.
11. Add another subwindow to the Waveform Display tool showing the passband results and make sure it is active.
12. Plot the I and Q baseband outputs from the baseband equivalent receiver model. Set the x-axis to be the I-output. You might need to adjust the scales on the last two plots to make them the same. Aside from the labels, the Waveform Display tool should look like Figure 1-110 on page 212.

© 1994-2020 All Rights Reserved.

Figure 1-110 Lissajous Plots for Baseband Signals



The baseband equivalent receiver model indeed rejects the image but the rejection is over-estimated. If you look closely, the baseband output of the passband model contains more ripple from the image. The over-attenuated ripple in the baseband model is explained as follows.

© 1994-2020
All Rights Reserved.

Recall the rotating reference frame analogy for baseband modeling. With respect to the rotating 1.1GHz reference frame, the image signals rotate counter-clockwise at twice the IF, 200MHz in this case. The lower left block in the baseband receiver model spins the modulator output at -200Mhz by ramping the phase error pin. The -200MHz signal propagates through the IF bandpass filters, as it should, because the response of the baseband model of the filter peaks at DC and at minus 200MHz. The trouble occurs in the final downconversion to baseband. In the baseband model, the final low pass filters severely attenuate the -200Mhz image signal. However, in the passband model, image power at minus 100MHz contributes to the baseband signal through the low pass filters with less attenuation.

This example highlights one of the limitations of baseband equivalent models: at any point in the system, the signal should not have a bandwidth larger than any carrier (RF or IF) of the system. For this example, the baseband model is only valid for input RF signals between 1GHz and 1.2GHz.

The limitation is somewhat moot because the idea behind a baseband equivalent model is to suppress all carriers. To simulate the image response with the baseband model we had to include a 200MHz source! We would have been better off simply not suppressing the 100MHz IF carrier, i.e. using baseband models for the RF stages but passband models for the IF stages.

In summary, an all-baseband equivalent model of an image rejection receiver is only good for simulating the response to the desired RF signal, not the image response.

© 1994-2020 All Rights Reserved.

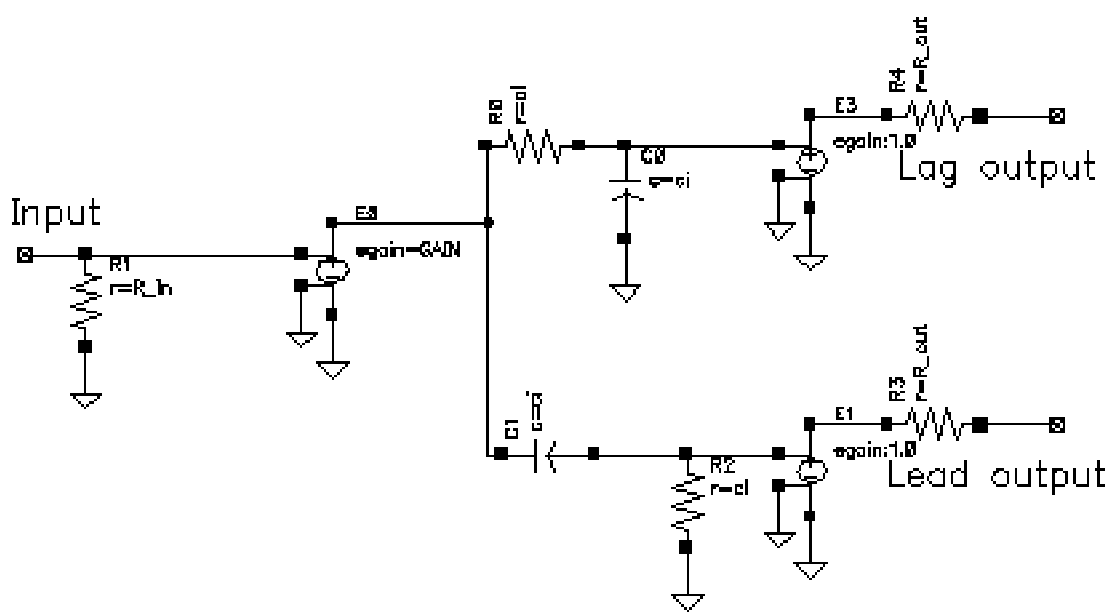
shifter_splitter_test

(Phase Shifter Splitter)

`(shifter-splitter)`

The phase shifter-splitter has one input and two outputs. The outputs are phase shifted versions of the input at the specified frequency. The phase difference between the two outputs is 90 degrees. The phase shifts are accomplished with Verilog-A code that does the same thing as the circuit shown in Figure 1-111 on page 214. The right-most voltage-controlled-voltage-sources (vcvs) are unity gain buffers. The left-most vcvs is also a buffer but the gain is a user-defined parameter. The input resistance, output resistance, intended operating frequency, and internal resistance are also user-defined. The internal resistance and operating frequency are used to calculate the capacitance necessary to provide +-45 degrees of phase shifts at the operating frequency. The baseband view requires the carrier frequency.

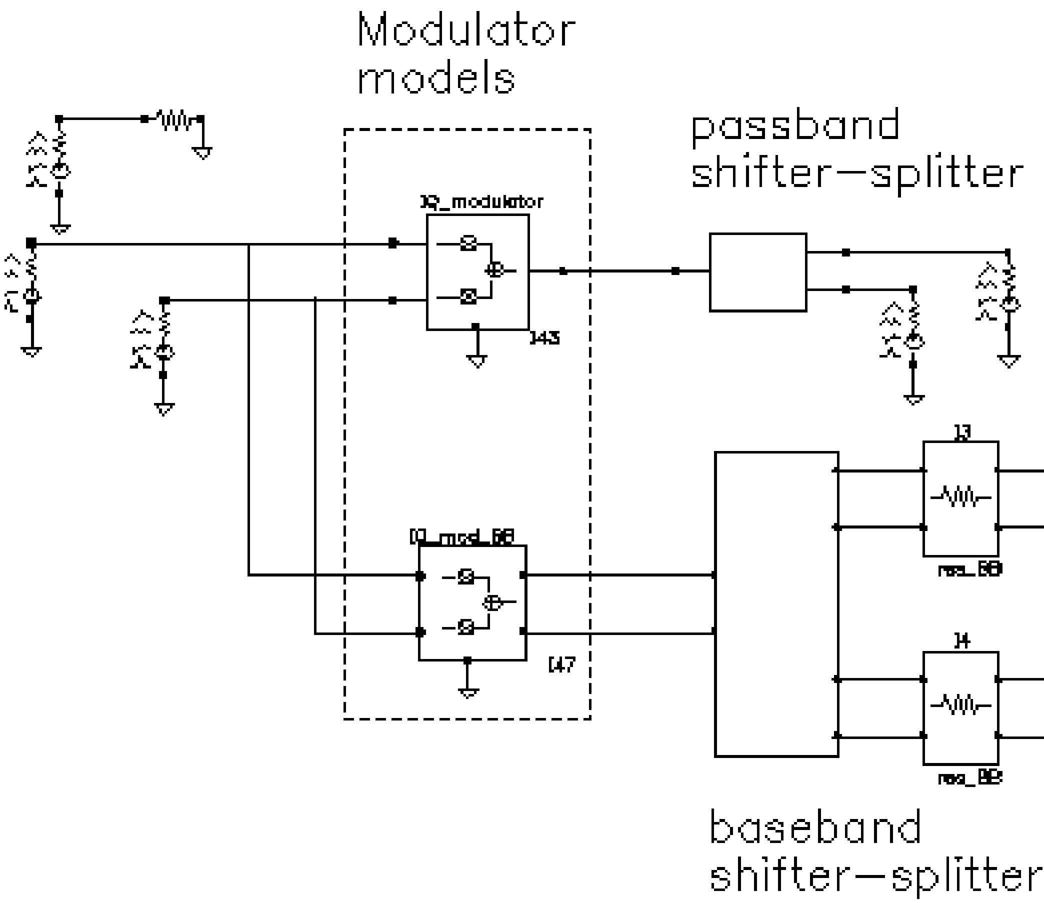
Figure 1-111 Phase Shift Circuit



The test circuit in Figure 1-112 on page 215 is for comparing the baseband responses of the passband and baseband equivalent models of the shifter-splitter. The circuit can be found in `rfLib` under the `testbenches` category. It is listed as `shifter_splitter_test`.

© 1994-2020 All Rights Reserved.

Figure 1-112 shifter_splitter_test Circuit

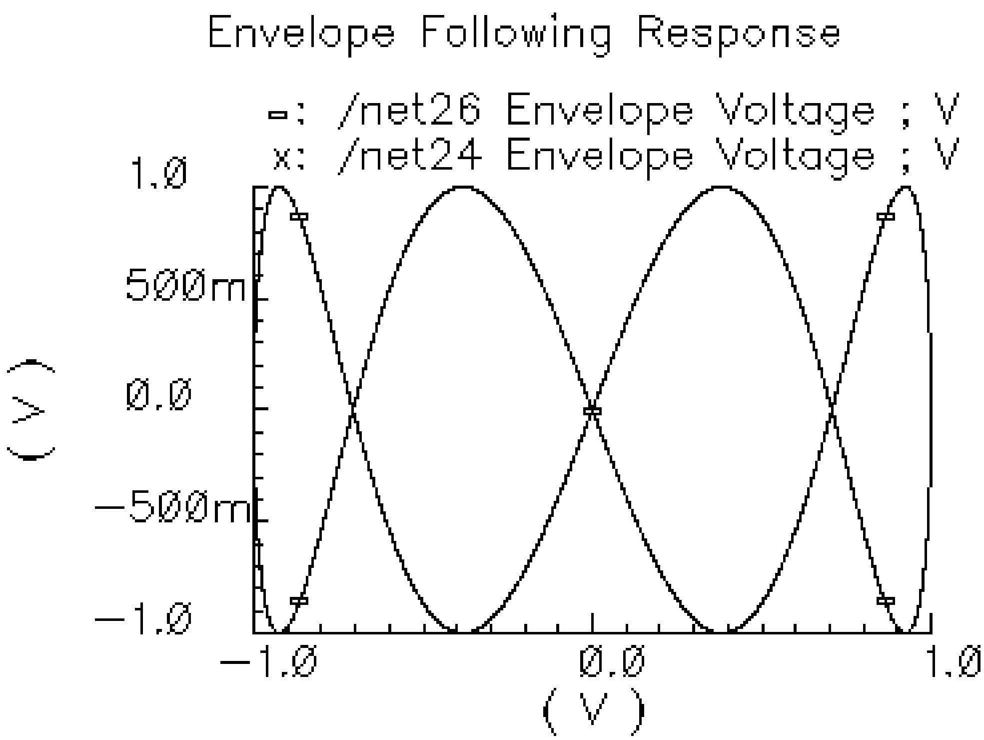


The following steps produce a set of Lissajous plots that show what the shifter-splitter does. You observe phase shift in the carrier by observing the tilt of the output Lissajous figures generated by the equivalent baseband signals.

1. Bring up the test circuit in Figure 1-112 on page 215 and an Analog Environment window.
2. Set up a 200ns Envelope analysis with `carrier` as the Clock Name. Set the Number of harmonics to 1. Set the Envelope analysis option called *modulationBW* equal to 100MHz.
3. Run the analysis.
4. Plot the `time` waveforms of the two input baseband signals. Change the x-axis to be the I-signal. You should see the Lissajous plot in Figure 1-113 on page 216.

© 1994-2020 All Rights Reserved.

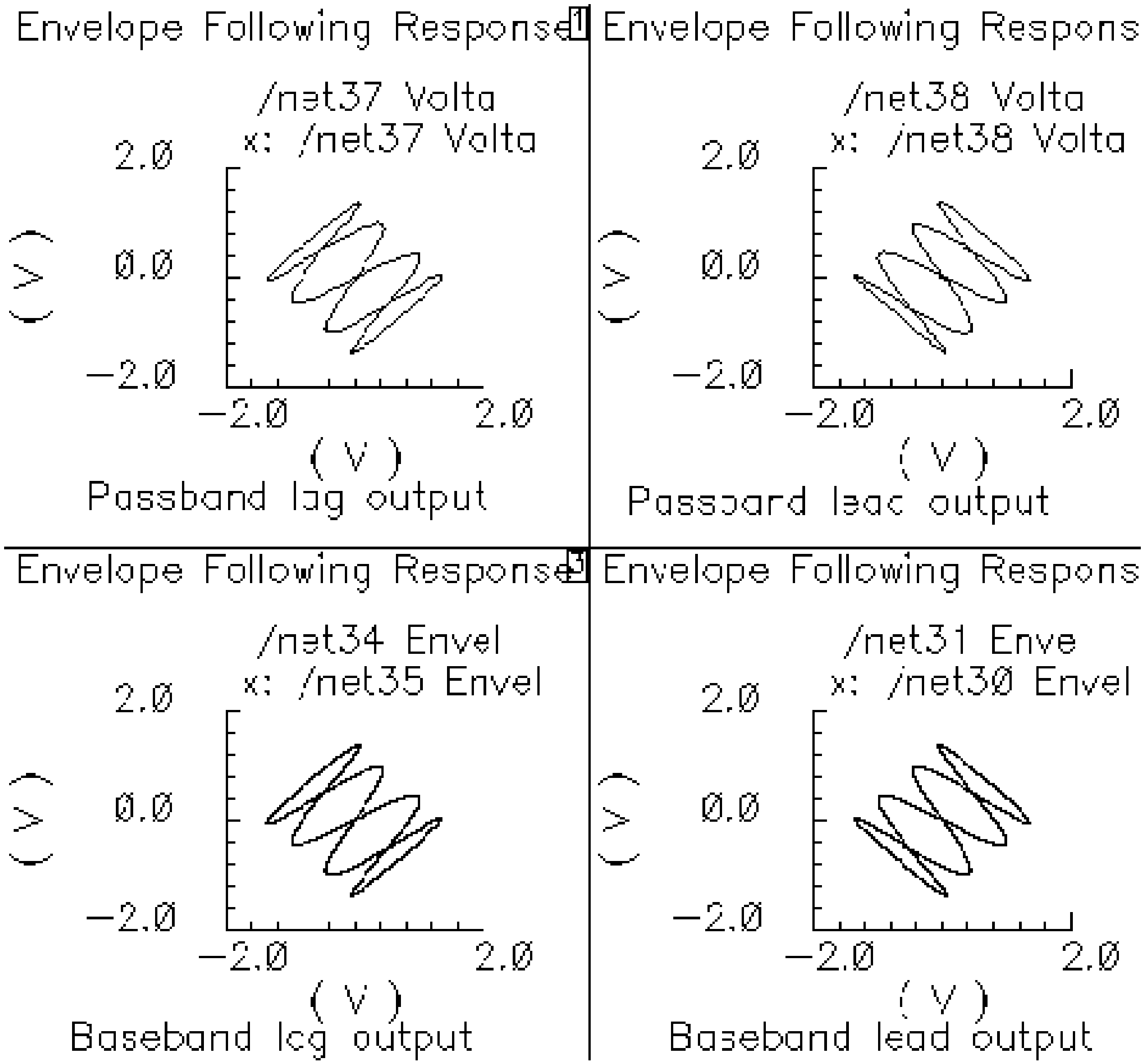
Figure 1-113 Lissajous Plot for Baseband Input Signals



5. Reset the Waveform Display window and plot the harmonic time, 1 harmonic, real and imaginary parts of the voltage across Port2. Set the x-axis to be the real part. Note that the Lissajous plot is tilted -45 degrees from the one in Figure 1-113 on page 216.
6. Add a subwindow.
7. Repeat step 5 for the voltage across Port1. Notice that the Lissajous plot is tilted +45 degrees with respect to the Lissajous plot in Figure 1-113 on page 216.
8. Add another subwindow.
9. Plot the `time` waveforms at the `lag_I` and `lag_Q` outputs of the `BB_shifter_splitter` model. Set the x-axis to be the lag_I waveform. The Lissajous plot should match the one produced in step 5.
10. Add another subwindow.
11. Repeat step 9 for the `lead` outputs of the `BB_shifter_splitter` model. The Lissajous plot should match the one produced in step 7. Aside from the labels, your Waveform Display tool should look like Figure 1-114 on page 217. The time-results of the baseband model faithfully duplicate the passband results but without simulating the carrier. The baseband model can be run with Spectre RF transient analysis.

© 1994-2020
All Rights Reserved.

Figure 1-114 Comparison of Lag and Lead times for Passband and Baseband Models



All Rights Reserved.

up_cnvt_test

(Testing the up_cnvrt Mixer)

There is a test circuit for the `up_cnvrt` model similar to the test circuit containing the `dwn_cnvrt` model. The `up_cnvrt` model is called `up_cnvt_test` and is shown in Figure 1-115. It is also in the `testbenches` category of `rfLib`. The steps parallel those for the `dwn_cnvrt` model.

Figure 1-115 up_cnvt_test Circuit

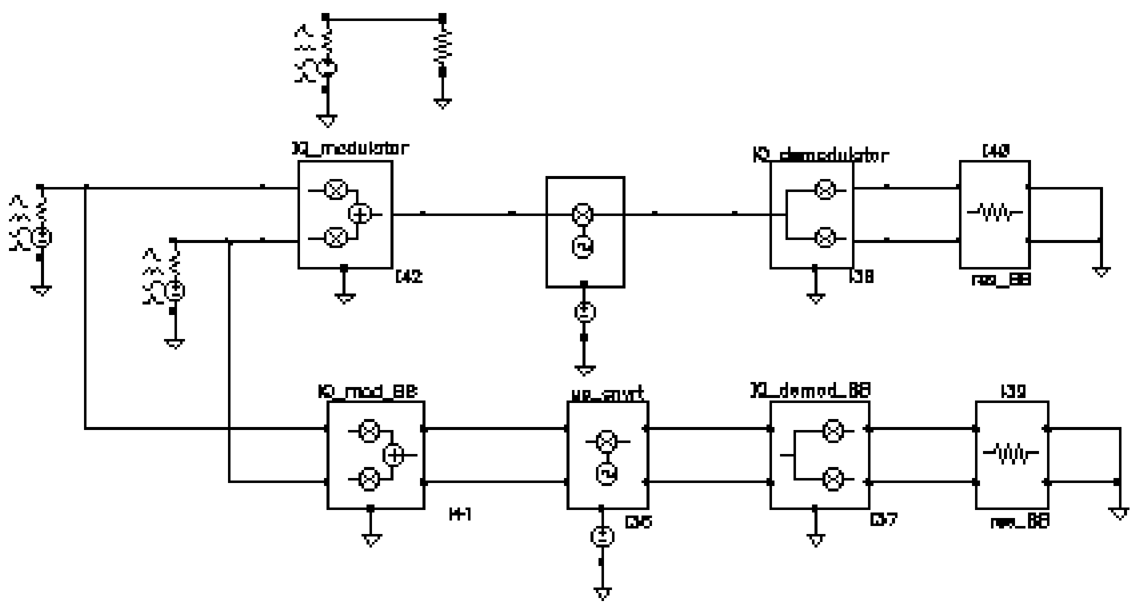
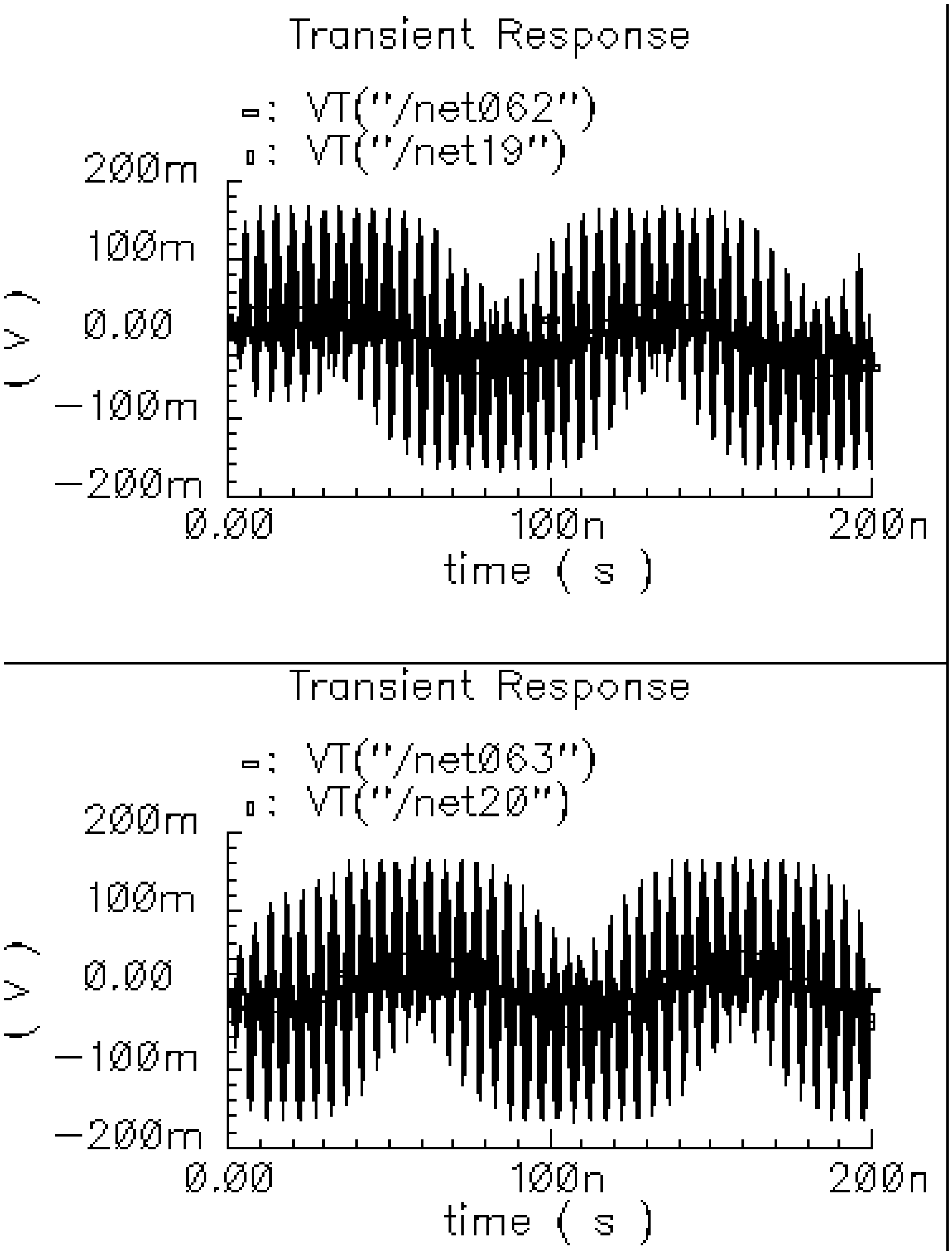


Figure 1-116 Results from a Transient Analysis



© 1994-2020 All Rights Reserved.

view_switching

This example illustrates how to switch between the single-ended baseband view and the differential passband view.

© 1994-2020
All Rights Reserved.

WCDMA_components Category

The components in the `WCDMA_components` category are:

- wcdma_dl_com_chanl
- wcdma_ocns
- wcdma_power_adjust
- wcdma_qpsk
- wcdma_sch_multiplexer
- wcdma_scrambling
- wcdma_scr_generator
- wcdma_spreading

© 1994-2020 All Rights Reserved.

wcdma_dl_com_chanl

(DL Common Channel Generator)

Figure 1-117 wcdma_dl_com_chanl symbol

This module, with four inputs and six outputs, generates the pCPICH (primary common pilot), PICH (paging indicator channel) and pCCPCH (primary common control physical channel). For PICH and pCCPCH, either external or internal random signals can be selected according to the value of the `enable_input` parameter. The parameters of the module are:

Name	Meaning	Type	Default Value	Range
`ccpch_seed`	Seed for CCPCH.	integer	98765	
`enable_input`	Enable input if 1; otherwise disable input.	integer	1	1, 0
`pich_seed`	Seed for PICH.	integer	12345	
`sample`	Sample time.	real	1.0/15000	

© 1994-2020 All Rights Reserved.

wcdma_ocns

(OCNS Generator)

Figure 1-118 wcdma_ocns symbol

This module produces the combination of 16 dedicated data channels. The parameters of the module are:

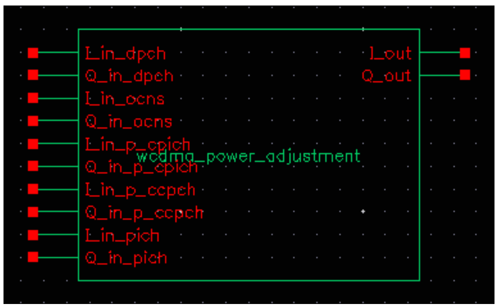
Name	Meaning	Type	Default Value	Range
`enable_input`	Enable input if 1, otherwise disable input.	integer	1	1, 0
`frame_time`	Frame time.	Real	1.0/15000	(0:inf)
`numChipsOut`	Number of chips.	integer	256	
`sf`	Spread factor.	integer	128	[4:512]

© 1994-2020
All Rights Reserved.

wcdma_power_adjust

(Power Adjustment)

Figure 1-119 wcdma_power_adjust symbol



This module applies the weight to different channels. The parameters of the module are:

Name	Meaning	Type	Default Value	Range
`frame_time`	Frame time.	real	1.0/15000	(0:inf)
`numChipsOut`	Number of chips.	integer	256	
`power_ccpch`	Power for CCPCH.	real	-12	
`power_cpich`	Power for CPICH.	real	-10	
`power_dpch`	Power for DPCH.	real	-5.5	
`power_pich`	Power for PICH.	real	-15	

© 1994-2020 All Rights Reserved.

wcdma_qpsk

(QPSK Modulation/Mapping)

Figure 1-120 wcdma_qpsk symbol

This block has two inputs, `I_in` and `Q_in`, and two outputs, `outi` and `outq`. The inputs receive random input from outside the module. The outputs produce QPSK signals in baseband.

- This module includes the interleaving and encoding of data.
- You can use either random input from the outside or internal random bits. If `enable_input` is `on`, external data is used, otherwise internal random bits are used.

The instance parameters for the `wcdma_qpsk` block are:

Name	Meaning	Type	Default Value	Range
`bits_per_integer`	Number of bits concerted into integer.	integer	2	[1:31]
`enable_input`	1 means to use outside input; otherwise 0.	Boolean	1	0, 1
`frame_time`	The time of one frame.	real	1.0/15000.0	(0:inf)
`mapping_mode`	Mapping type.	string	user_defined	Binary_gray, gray_binary, user_defined
`phase_offset`	Initial phase.	real	0	
`samples`	Number of samples in one frame.	integer	12345	

© 1994-2020
All Rights Reserved.

`seed`	Used for random.	type	2	
`usr_mapping_vec`	Bit mapping between input and output.	integer	{0, 3, 1, 2}	

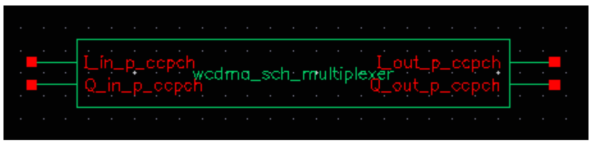
© 1994-2020

All Rights Reserved.

wcdma_sch_multiplexer

(SCH Generator/Multiplexer)

Figure 1-121 wcdma_sch_multiplexer symbol



This module produces the synchronization channel and performs SCH multiplexing. The parameters of the module are:

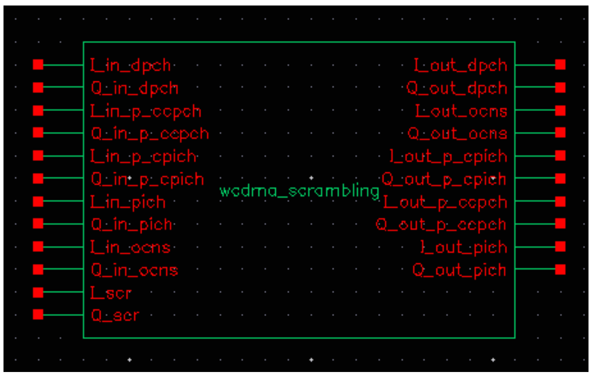
Name	Meaning	Type	Default Value	Range
`frame_time`	Frame time	real	1.0/15000	(0:inf)
`hada_order`	Hadamard matrix order.	integer	8	
`numChipsOut`	Number of chips.	integer	256	
`ssc_num`	Scrambling code group number.	integer	64	[1:64]

© 1994-2020 All Rights Reserved.

wcdma_scrambling

(Scrambling/Scrambling Code)

Figure 1-122 wcdma_scrambling symbol



This module scrambles the spread code. The parameters of the `wcdma_scrambling` module are:

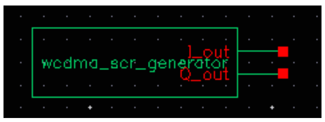
Name	Meaning	Type	Default Value	Range
`frame_time`	Frame time.	real	1.0/15000	(0:inf)
`numChipsOut`	Number of chips.	integer	256	

© 1994-2020 All Rights Reserved.

wcdma_scr_generator

(Square-Root Raised Cosine)

Figure 1-123 wcdma_scr_generator symbol



This module up-samples and filters the input. The parameters of the module are:

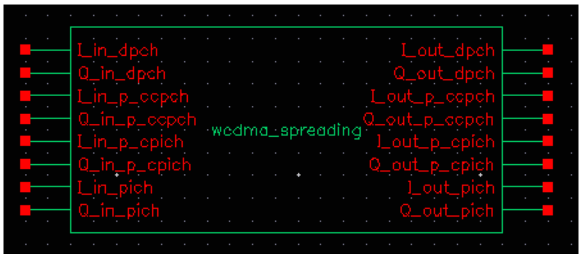
Name	Meaning	Type	Default Value	Range
`alpha`	Filter attenuation at cutoff [dB].	real	0.22	
`frame_time`	Frame time.	real	1.0/15000	(0:inf)
`group_delay`		integer	6	
`numChipsOut`	Number of chips.	integer	256	
`over_samples`		integer	8	

© 1994-2020 All Rights Reserved.

wcdma_spreading

(Spreading)

Figure 1-124 wcdma_spreading symbol



This module spreads the data over the OVSF codes. The parameters of the `wcdma_spreading` module are:

Name	Meaning	Type	Default Value	Range
`dpch_code`	OVSF index for DPCH.	integer	10	
`frame_time`	Frame time.	real	1.0/15000	
`oversample`	Oversample.	integer	128	
`pccpch_code`	OVSF index for PCCPCH.	integer	1	
`pcpich_code`	OVSF index for PCPICH.	integer	0	
`pich_code`	OVSF index for PICH.	integer	4	
`sf`	Spread factor.	integer	128	[4:512]

© 1994-2020 All Rights Reserved.

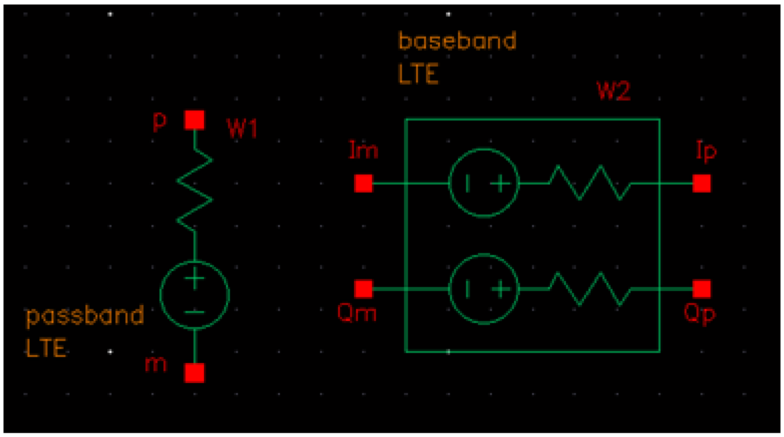
Wireless Components Category

Wireless components automate the measurements associated with modulated signals. The supported standards are Zigbee, 802.11a, 802.11n, 802.11ac, 802.11ad, 802.11af, 802.11ah, 802.11p, 802.11ax, Bluetooth (BR, EDR, LE, and HS modes), smart meter, dmr, LTE, and NR. To use wireless, add a wireless source to your circuit, and add wireless probes at each point in the circuit where you want a measurement. When you set up the envelope analysis, put envelope into wireless mode, and specify the number of harmonics and whether you want standard envelope or level1 fast envelope. Run the simulation. In the Direct Plot Form, all the common measurements are available directly. This is much easier than in the past.

© 1994-2020 All Rights Reserved.

LTE

SYMBOL



LTE Modulation Source

Command-line Help

```
spectre -h wsource
```

CDF Parameters

Label	Parameter	Type	Default
Band Type	N/A	Enum	Passband
Average Power (dbm)	dbm	Double	6.99
Resistance	R	Double	50
Num of SF (FDD)/HF (TDD)	framenumber	Integer	1
Oversample ratio	oversample	Integer	4
SNR (db)	snr	Double	N/A (infinite)
Operating Band	LTE_Band	Integer	1
FDD or TDD Mode	LTE_Mode	Enum	TDD
Direction	LTE_Direction	Enum	UL

© 1994-2020 All Rights Reserved.

Channel Number	LTE_ChannelNumber	Integer	18050
Channel bandwidth	LTE_Bandwidth	Enum	10M
Modulation Type	LTE_Modulation	Enum	QAM16
Channel offset	LTE_ChannelOffset	Integer	0
Subcarrier spacing	LTE_DeltaFUL	Enum	15K
Number of tone	LTE_ToneNumUL	Integer	1
Tone start index	LTE_ToneStartIndex UL	Integer	0
Resource Block Allocation	LTE_RBAllocation	Enum	Full
Resource Block Number	LTE_RBNumber	Integer	12
Starting Index of RB	LTE_RBStartIndex	Integer	0
Number of carriers in CA	LTE_CANum	Integer	1
ACPR Type or CIM	LTE_ACPRType	Enum	EUTRA
Downlink signal type	LTE_SignalTypeDL	Enum	PDSCH

Parameter Description

- *Band Type*: sets the passband or baseband operating mode.
- *Average Power*: source available average power in dBm
- *Resistance*: source Thevenin resistance
- *Number of SF (FDD)/HF (TDD)*: The number of subframes to simulate.
- *Oversample ratio*: Controls the sampling frequency. If *ACPR Type* is set to *UTRA and Channel bandwidth* is specified as 200k, the value should be greater than or equal to 3. The sampling frequency is given by oversample*32.72MHz.
- *SNR*: Signal-to-noise power ratio.
- *Operating band*: Operating band given by TS36.101 Table 5.5-1, in the set {[1:14], [17:28], [30:31], [33-48], 65-66], [68], [70]}. Band [33-48] is TDD mode while the others are FDD mode.
- *FDD or TDD Mode*: Selects the FDD or TDD Mode. In the default *TDD* mode, two out of five subframes are populated. In *FDD* mode, all subframes are populated. Possible values are `FDD` and `TDD`. This option is displayed only when the operating band is in the range [33:48].

© 1994-2020 All Rights Reserved.

- *Direction*: Direction of uplink or downlink. Possible values are UL and DL.
- *Channel number*: E-UTRA Absolute Radio Frequency Channel Number given by TS36.101 Table 5.7.3-1, in the set {(18000:27809), (36000:55239), (131072:133221)}.
- *Channel bandwidth*: E-UTRA channel bandwidth given by TS36.101 Table 5.6.1-1, in the set {200K, 1.4M, 3M, 5M, 10M, 15M, 20M}.
- *Modulation type*: Specifies the modulation type. For LTE, it is QPSK, QAM16, and QAM64. For NB1 uplink, it is QPSK and BPSK. NB1 downlink always uses QPSK. Possible values are QPSK, QAM16, QAM64, and BPSK.
- *Channel offset*: Specifies the offset from the NB1 channel number center frequency. For UL, it is in the set {-10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9}. For DL, it is in the set {-10, -9, -8, -7, -6, -5, -4, -3, -2, -1, -0.5, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9}.
- *Subcarrier spacing*: Specifies the subcarrier spacing for NB1 uplink in the set {15K, 3.75K}.
- *Number of tone*: Specifies the number of tones for NB1 uplink category. For 15K subcarrier spacing, it is in the set {1, 3, 6, 12}. For 3.75K subcarrier spacing, it is 1.
- *Tone start index*: Specifies the tone start index for the NB1 uplink category.
- *Resource Block Allocation*: Choices are `Full` and `Partial`. When `Full` is selected, all the resource blocks in the channel are fully populated. When `Partial` is selected, the number of resource blocks and their location within the channel must be set.
- *Resource Block Number*: This sets the number of resource blocks that are occupied in the channel. The number should be 2^m * 3^n * 5^p where m, n, p >= 0. Each resource block occupies 180KHz bandwidth. The maximum number depends on the channel bandwidth. The maximum is shown in the table below.

Parameter	Unit	Value					
Channel Bandwidth	MHz	1.4	3	5	10	15	20
Maximum Number of Resource Blocks		6	15	25	50	75	100

- *Starting Index of RB*: This sets the starting point in the channel for the resource blocks. Zero means start at the lower boundary of the channel. The starting index + number of resource blocks must be equal to or less than the maximum number shown in the table above.
- *Number of carriers in CA*: This sets the number of carriers intra band channel aggregation. The default is 1. Values from 1 through 5 are selectable. Note that when set

© 1994-2020
All Rights Reserved.

from 2 through 5, the EVM plot shows considerably larger values than are real. This option is available only when Channel bandwidth is set to 3M, 5M, 10M, 15M, or 20M.

- *ACPR Type or CIM*: This is selectable between EUTRA, which is the default, UTRA, and CIM. These settings are defined in 36101 section 6.6.2.3.1 for EUTRA and 6.6.2.3.2 for UTRA.
- *Downlink signal type*: Specifies the downlink signal type. When set to *PDSCH&CRS*, both PDSCH and CRS are transmitted. When set to *PDSCH*, only PDSCH is transmitted. When set to *CRS*, only CRS is transmitted. Possible values are: *PDSCH&CRS*, *PDSCH*, and *CRS*.

  Note: This option is available when *DL* is selected for *Direction*.

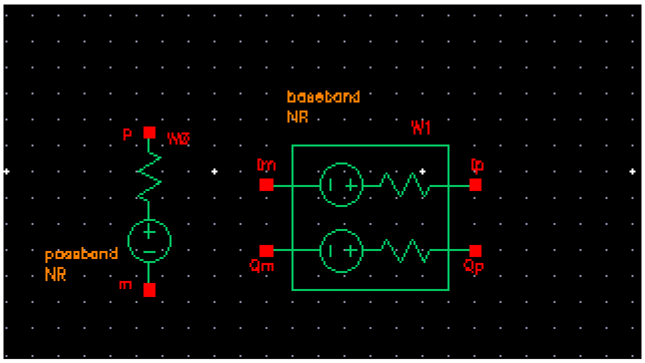
Example

```
W0 ( net2 net1  ) wsource standard="LTE" r=50 dbm=10  LTE_Band=1
LTE_ChannelNumber=18050 LTE_Bandwidth=10M LTE_Modulation=QAM16  framenumber=1
oversample=4
```


© 1994-2020 All Rights Reserved.

NR

SYMBOL



NR Modulation Source

Command-line Help

```
spectre -h wsource
```

CDF Parameters

Label	Parameter	Type	Default
Band type	N/A	Enum	Passband
Average Power (dbm)	dbm	Double	6.99
Resistance	R	Double	50
Num of half subframe	framenumber	Integer	1
Oversample ratio	oversample	Integer	4
SNR(dB)	snr	Double	N/A (infinite)
Operating band	NR_Band	Integer	1
Direction	NR_Direction	Enum	UL
OFDM mode	NR_OFDMMode	Enum	CP
Channel number	NR_ChannelNumber	Integer	384000

© 1994-2020
All Rights Reserved.

Channel bandwidth	NR_Bandwidth	Enum	10M
Modulation type	NR_Modulation	Enum	QAM16
Subcarrier spacing	NR_SCS	Enum	15K
RB number	NR_RBNumber	Integer	50
RB start index	NR_RBStartIndex	Integer	0
Number of carriers in CA	NR_CANum	Enum	1
PAPR Reduction in dB	NR_CFR	Integer	0
EVM threshold specified	NR_EVM	Double	.03

Parameter Description

- *Band type*: sets the passband or baseband operating mode.
- *Average Power*: source available average power in dBm.
- *Resistance*: source Thevenin resistance.
- *Num of half subframes*: The number of subframes to simulate.
- *Oversample ratio*: Controls the sampling frequency. The sampling frequency is given by oversample*Subcarrier spacing*4096.
- *SNR*: Signal-to-noise power ratio.
- *Operating band*: Operating band given by TS38.101 Table 5.2-1. For `UL`, it is in the set {[1:3], 5, [7:8], 12, 20, 25, 28, 34, [38:41], [50:51], 66, [70:71], 74, [77:84], 86, [257:258], [260:261]}. For `DL`, it is in the set {[1:3], 5, [7:8], 12, 20, 25, 28, 34, [38:41], [50:51], 66, [70:71], [74:79], [257:258], [260:261]}.
- *Direction*: Direction of uplink or downlink. Possible values are *UL* and *DL*.
- *OFDM_mode*: The OFDM mode for uplink (`UL`). Possible values are *CP* and *DFT*.
- *Channel number*: Absolute Radio Frequency Channel Number given by TS38.101 Table 5.4.2.3-1, in the range [0: 2279165].
- *Channel bandwidth*: Channel bandwidth in the range [5M, 10M, 15M, 20M, 25M, 30M, 40M, 50M, 60M, 80M, 90M, 100M, 200M, 400M].
- *Modulation type*: Specifies the modulation type. Possible values are *QPSK*, *QAM16*, *QAM64*, *QAM256* for direction *DL* and *UL* (with *OFDM mode* set to *CP*), and *QPSK*, *QAM16*, QAM64, *QAM256*, and *pi_2_BPSK* for *UL* with *OFDM mode* set to *DFT*.

© 1994-2020 All Rights Reserved.

- *Subcarrier spacing*: Specifies the subcarrier spacing in the set {15K, 30K, 60K, 120K}.
- *RB number*: This sets the number of resource blocks that are occupied in the channel. For UL, the number should be 2^m * 3^n * 5^p where m, n, p >= 0.
- *RB start index*: This sets the starting point in the channel for the resource blocks.
- *Number of carriers in CA*: This sets the number of carriers intra band channel aggregation. The default is 1. Values from 1 through 5 are selectable. Note that when set from 2 through 5, the *RB number* and *RB start index* options are ignored. This option is visible only when the *Operating band* is specified as `77`, `78`, `79`, `257`, `260`, and `261`.
- *PAPR reduction in dB*: Specifies the peak-to-average power ratio (PAPR) reduction (in dB) in the range [-6:0].
- *EVM threshold specified*: Specifies the EVM threshold in the range [0:0.5].

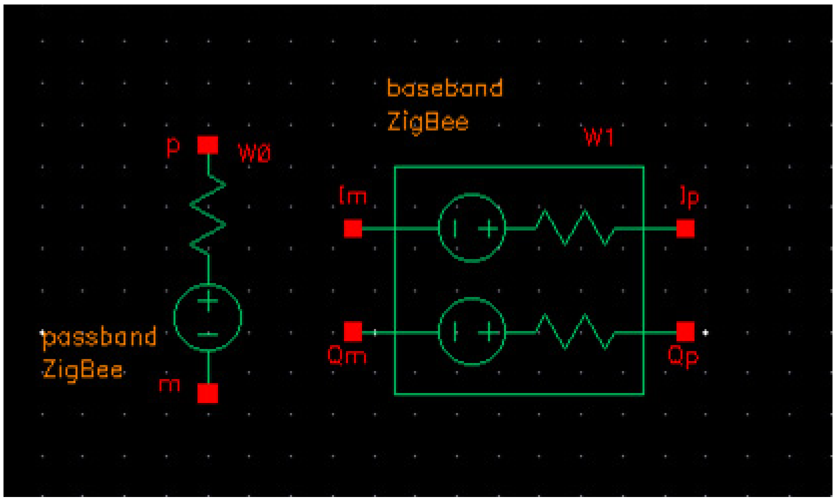
Example

```
W0 (net6 0 net3 0) wsource standard="NR" dbm=6.99 r=50 framenumber=1 \
 oversample=8 snr=40 NR_Band=79 NR_Direction=UL NR_OFDMMode=CP \
 NR_ChannelNumber=733333 NR_Bandwidth=100M NR_SCS=60K \
 NR_Modulation=QAM16 NR_CANum=4
```


© 1994-2020 All Rights Reserved.

ZIGBEE

SYMBOL



ZigBee modulation source

Command-line help

```
spectre -h wsource
```

Notes

`zigbee` applies ZigBee modulation compliant with *IEEE Std 802.15.4-2006, Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*. The present implementation is limited to the 2400 MHz band. `zigbee` netlists as a Spectre `wsource`.

Frequency Band

The 2400 MHz band consists of 16 channels spaced by 5 MHz. The carrier frequency is given by `Fc = 2405 + 5*(k-11) MHz`.

`k` is the channel index in the range `[11:26]`. The parameter *Channel* determines the value of `k`.

© 1994-2020
All Rights Reserved.

You can reference `Fc` in the schematic and analysis dialogs as `<instance_name>_wfreq`, where `<instance_name>` is the schematic instance name of the *zigbee* cell.

Frame structure

A ZigBee frame consists of `HeaderLength + DataLength` octets (bytes), encoded in `(HeaderLength + DataLength)*64` chips and transmitted at 2M chips/s. The duration of a frame is therefore `Tframe = (HeaderLength + DataLength)*32 us`.

Sampling frequency

The sampling frequency in the 2405M band is given by `Fs = 4M*Oversample`.

`Oversample` is a user-settable parameter.

CDF Parameters

Label	Parameter	Type	Default
Band Type	N/A	Enum	Passband
Power (dBm)	dbm	Double	6.99
Resistance	R	Double	50
Number of Frames	framenumber	Integer	1
Oversample ratio	oversample	Integer	3
Channel	ZigBee_Channel	Integer	11
PN code	ZigBee_DataType	String	PN15
Data length	ZigBee_DataLength	Integer	20
SNR (db)	snr	Double	N/A (infinite)

Parameter Descriptions

- *Band type*: sets the passband or baseband operating mode.
- *Power*: source available average power in dBm.
- *Resistance*: source Thevenin resistance.

© 1994-2020 All Rights Reserved.

- *Number of frames*: The number of frames to simulate. The length of a frame is `(6 + Data length)*32u`.
- *Channel*: Channel index `k`. The channel index controls the carrier frequency, which is given by `Fc = 2405 + 5*(Channel-11) MHz`.
- *PN code*: The pseudo-random sequence that controls the data patterns.
- *Data length*: The number of data octets/bytes in the range [0:127]. Data length controls the length of a frame.
- *Oversample ratio*: Controls the sampling frequency. The sampling frequency is given by `Oversample*4MHz`.
- *SNR*: Signal-to-noise power ratio.

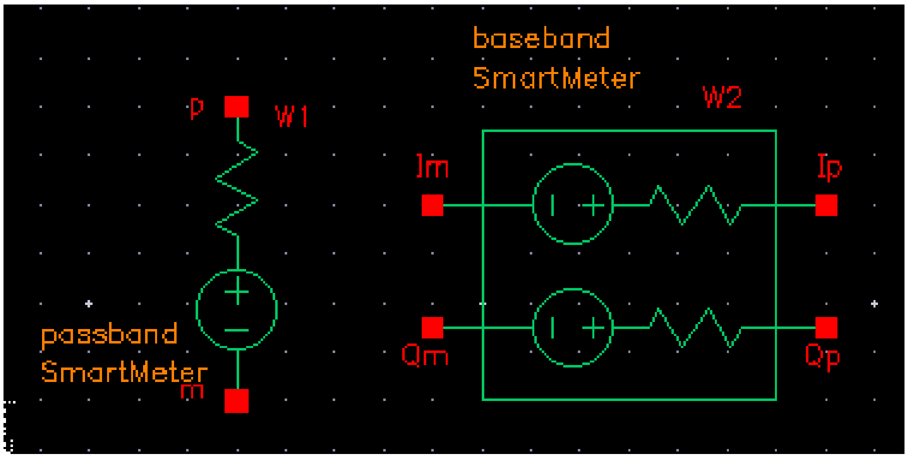
Example

```
W0 (net2 net1) wsource standard="ZigBee" r=50 dbm=6.99 ZigBee_Channel=11 \
ZigBee_DataType=PN15 ZigBee_DataLength=20 oversample=3 framenumber=2
```


© 1994-2020 All Rights Reserved.

smartmeter

SYMBOL



smartmeter modulation source

Command-line help

```
spectre -h wsource
```

Notes

`smartmeter` applies modulation compliant with *IEEE Std 802.15.4g-2012.* `smartmeter` netlists as a Spectre `wsource`.

`smartmeter` supports FSK Mode 1, OFDM Option 4, and OQPSK RateMode 3.

Frequency Band

All frequency bands 0 through 13 as defined in Table 68f are supported.

Frame structure

A `smart_meter` frame consists of a header and data that depends on the modulation type, as set by the PHY Type parameter.

© 1994-2020 All Rights Reserved.

CDF Parameters For All Modulation Types

Label	**Parameter**	**Type**	**Default**
Band Type	N/A	Enum	Passband
Power (dBm)	dbm	Double	6.99
Resistance	R	Double	50
Number of Frames	framenumber	Integer	3
SNR (dB)	snr	Double	Unspecified (Infinite)
Oversample ratio	oversample	Integer	4
PN code	SmartMeter_DataType	String	PN9

Parameter Descriptions For All Modulation Types

- *Band type*: sets the passband or baseband operating mode.
- *Power*: source available average power in dBm.
- *Resistance*: source Thevenin resistance.
- *Number of frames*: Number of frames to simulate.
- *PN code*: Sets the length of the shift register for the pseudo-random sequence that generates the data patterns.
- *Oversample ratio*: Controls the sampling frequency. Use the default of 4 for most simulations. If exceptionally wide bandwidth is needed in the PSD, set the parameter to 8.
- *SNR*: Signal-to-noise power ratio.

CDF Parameters for FSK

Label	**Parameter**	**Type**	**Default**
Frequency band identifier	SmartMeter_Band	Integer	0
Channel number	SmartMeter_Channel	Integer	0
Preamble length in octets	SmartMeter_PreambleLength	Integer	4

© 1994-2020 All Rights Reserved.

Payload length in octets	SmartMeter_DataLength	Integer	1000

Parameter Descriptions for FSK

- *Frequency band identifier:* Sets the frequency band number from table 68f. Possible values for FSK are 0 through 13.
- *Channel number:* Channel numbers as defined in Table 68d are supported.
- *Preamble length in octets:* Preamble length as defined in Table 71. As per the table, values from 4 through 1000 are supported.
- *Payload length in octets:* Size of the data portion in octets. Values from 1 through 2047 are supported.

CDF Parameters for OFDM

Label	Parameter	Type	Default
Frequency band identifier	SmartMeter_Band	Integer	2
Channel number	SmartMeter_Channel	Integer	0
Number of data symbols	SmartMeter_DataSym	Integer	80
Modulation and coding	SmartMeter_MCS	Integer	2
EVM measurement type	SmartMeter_EVMType	String	Data&Pilot

Parameter Descriptions for OFDM

- *Frequency band identifier:* Sets the frequency band number from table 68f. Possible values for OFDM are 2, 3, 4, 7, 8, 9, 11, or 13.
- *Channel number:* Channel numbers as defined in Table 68d are supported.
- *Number of data symbols:* Number of OFDM data symbols in one frame.
- *Modulation and coding:* MCS 2 through 6 are implemented. Only OFDM Option 4 is supported.
- *EVM measurement type:* Values can be Data&Pilot, Data, or Pilot.

© 1994-2020 All Rights Reserved.

CDF Parameters for OQPSK

Label	Parameter	Type	Default
Frequency band identifier	SmartMeter_Band	Integer	13
Channel number	SmartMeter_Channel	Integer	0
Payload length in octets	SmartMeter_DataLength	Integer	1000

Parameter Descriptions for OQPSK

- *Frequency band identifier:* Sets the frequency band number from table 68f. The only possible value for OQPSK is 13.
- *Channel number:* Channel numbers as defined in Table 68d are supported.
- *Payload length in octets:* Size of the data portion in octets.

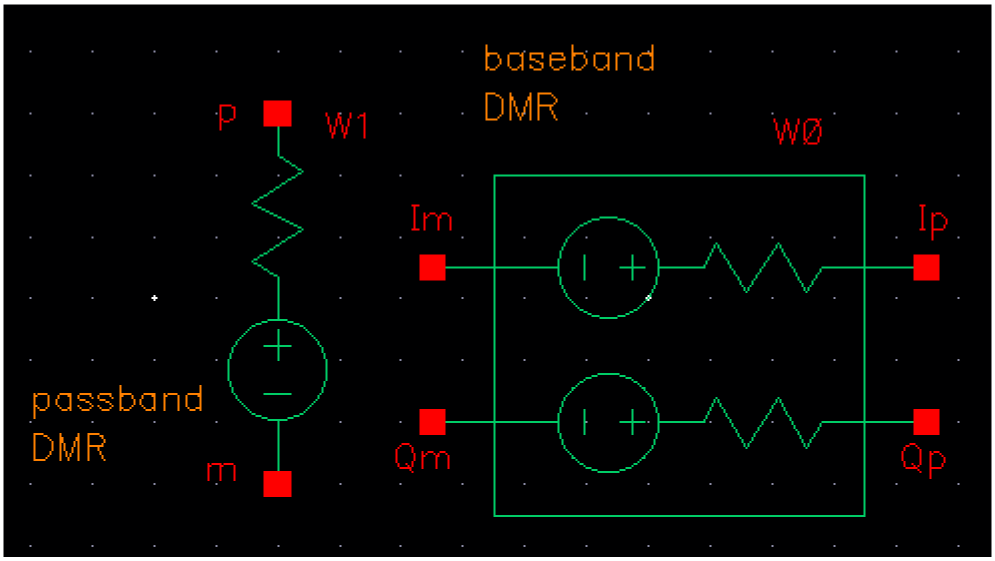
Examples

```
FSK (net2 0) wsource standard="SmartMeter" r=50 dbm=6.99 framenumber=3 \
oversample=4 SmartMeter_Channel=0 SmartMeter_DataType=PN9 \
SmartMeter_PHYType=FSK \
SmartMeter_Band=0 SmartMeter_PreambleLength=4 \
SmartMeter_DataLength=1000

OFDM (net2 0) wsource standard="SmartMeter" r=50 dbm=6.99 framenumber=3 \
oversample=4 SmartMeter_Channel=0 SmartMeter_DataType=PN9 \
SmartMeter_PHYType=OFDM \
SmartMeter_Band=2 SmartMeter_DataSym=80 SmartMeter_MCS=2 \
SmartMeter_EVMType=Data&Pilot

OQPSK (net2 0) wsource standard="SmartMeter" r=50 dbm=6.99 framenumber=3 \
oversample=4 SmartMeter_Channel=0 SmartMeter_DataType=PN9 \
SmartMeter_PHYType=OQPSK \
SmartMeter_Band=13 SmartMeter_DataLength=1000
```


© 1994-2020 All Rights Reserved.

DMR

Symbol



DMR Modulation Source

Command-line Help

```
spectre -h wsource
```

Notes

`DMR` netlists as a Spectre `wsource`.

Frequency Band

The carrier frequency is in the 30MHz to 1GHz band. As always, you can reference the carrier frequency in the schematic and analysis dialogs as `<`*`instance_name`*`>_wfreq`, where `<`*`instance_name`*`>` is the schematic instance name of the *DMR* cell.

Frequency Bandwidth

DMR operates within a 12.5 KHz RF carrier bandwidth.

© 1994-2020
All Rights Reserved.

CDF Parameters

Label	**Parameter**	**Type**	**Default**
Band type	`N/A`	Enum	Passband
Average Power	`dBm`	Double	6.99
Resistance	`r`	Double	50
Number of bursts	`framenumber`	Integer	1
Oversample ratio	`oversample`	Integer	8
SNR(db)	`snr`	Double	N/A (Infinite)
Carrier frequency	`DMR_Frequency`	String	400M
PN Code	`DMR_DataType`	Enum	PN9
Pulse shaping filter type	`DMR_FilterType`	Enum	RRC
Roll-off factor	`DMR_Alpha`	String	0.2
Idle time	`DMR_IdleTime`	Enum	0

Parameter Descriptions

- *Band type*: Sets the passband or baseband operating mode.
- *Average Power*: Source available average power in dBm.
- *Resistance*: source Thevenin resistance in Ohms.
- *Number of bursts*: The number of frames to simulate. Default is 1.
- *Oversample ratio*: Controls the sampling frequency. The default is 4.
- *SNR*: Signal-to-noise power ratio.
- *Carrier frequency*: Channel center frequency, in range [30M:1G].
- PN Code: The pseudo random sequence for the transmitted data. Possible values are *PN9*, *PN15*, and *PN23*.
- *Pulse shaping filter type*: Pulse shaping filter type. Possible values are *None* and *RRC*.
- *Roll-off factor*: Roll-off factor, in range (0:1).

© 1994-2020 All Rights Reserved.

- *Idle time*: Idle time in a TDMA frame. Possible values are 0 and 30m.

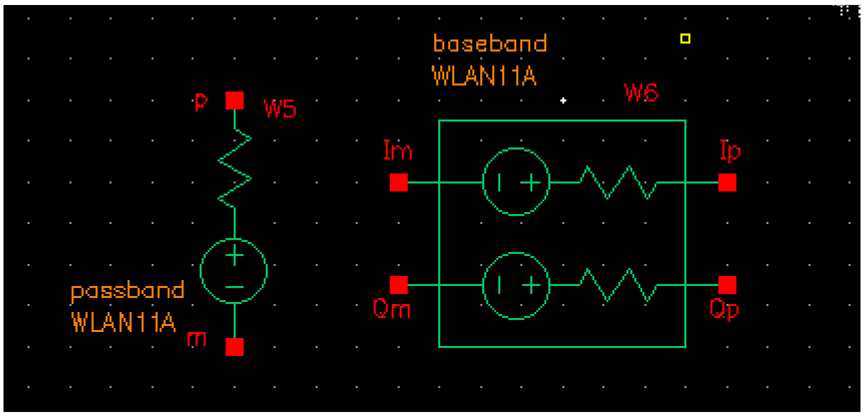
Example

```
W0 (net2 net1) wsource standard="DMR" r=50 dbm=6.99 DMR_Frequency=200M
DMR_Alpha=0.1 DMR_IdleTime=30M
```


© 1994-2020 All Rights Reserved.

WLAN11A

SYMBOL



802.11a modulation source

Command-line help

`spectre -h wsource`

Notes

`wlan11a` applies 802.11a-compliant modulation according to *IEEE Standard 802.11a-1999, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, High-Speed physical Layer in the 5 GHz Band*.

`wlan11a` netlists as a Spectre `wsource`.

Frequency Band

The carrier frequency is in the 5 GHz band. As always, you can reference the carrier frequency in the schematic and analysis dialogs as `<`*`instance_name`*`>_wfreq`, where `<`*`instance_name`*`>` is the schematic instance name of the *wlan11a* cell.

© 1994-2020 All Rights Reserved.

Frame structure

A `wlan11a` frame consists of preamble and data. Data length is a settable property with a default of 20 symbols. The frame duration and sampling frequencies are given in the table below assuming 20 data symbols.

	Preamble (time)	Data (time for 20 symbols)	Total time	fftlength	Time Step
20M	20u	80u	Preamble+ Data	Oversample* 64	1/20M/Oversa mple

	Constellation	Data Carriers (of 1 symbol)	Pilots (of 1 symbol)
20M	64-QAM	48	4

CDF Parameters

Label	**Parameter**	**Type**	**Default**
Band Type	N/A	Enum	Baseband
Average Power (dbm)	dbm	Double	6.99
Resistance	r	Double	50
Number of Frames	framenumber	Double	1
Oversample ratio	oversample	Integer	4
SNR (db)	snr	Double	N/A (Infinite)

© 1994-2020 All Rights Reserved.

Channel Number	WLAN11N_Channel	Integer	0
Number of Data Symbols	WLAN11N_DataSym	Integer	20
EVM	WLAN11N_EVMType	Enum	Data&Pilot

Note: 802.11a is 802.11n in the 5GHz frequency band with the long guard interval using legacy format and fixed bandwidth of 20MHz. 802.11a netlists as 802.11n with the appropriate settings.

Parameter Descriptions

- *Band type*: Sets the passband or baseband operating mode.
- *Average Power*: Source available average power in dBm.
- *Resistance*: source Thevenin resistance in Ohms.
- *Number of frames*: The number of frames to simulate. Refer to the table for the frame *Oversample*: Controls the sampling frequency. Refer to the table for details.
- Oversample: Controls the sampling frequency. The default is 4.
- *SNR*: Signal-to-noise power ratio.
- *Channel number*: Channel number in range [0:200] in the 5GHz band. The carrier frequency is given by `Fc (MHz) = 5000 + 5*WLAN11N_Channel`.
- *EVM measurement type*: EVM measurement type. When set to `Data&Pilot`, the EVM calculation includes all the used subcarriers of Data and Pilots. When set to `Data`, the EVM calculation includes only the Data subcarriers. When set to `Pilot`, the EVM calculation includes only the Pilot subcarriers. Possible values are `Data&Pilot`, `Data`, and `Pilot`.

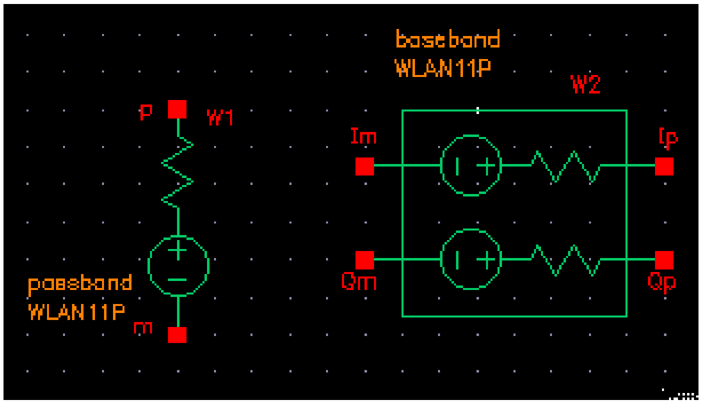
Example

```
W0 (net2 net1) wsource standard="WLAN11N" r=50 dbm=6.99 WLAN11N_Mode=Legacy \
WLAN11N_Bandwidth=20M WLAN11N_Oversample=4 WLAN11N_GI=Long framenumber=1
WLAN_11N_Channel=100 WLAN_11NDataSym=20
```


© 1994-2020 All Rights Reserved.

WLAN11P

SYMBOL



802.11p modulation source

Command-line help

`spectre -h wsource`

Notes

`wlan11p` applies 802.11p-compliant modulation according to *IEEE Standard 802.11p-2010, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications*.

`wlan11p` netlists as a Spectre `wsource`.

Frequency Band

The carrier frequency is in the 5 GHz or 700 MHz band. As always, you can reference the carrier frequency in the schematic and analysis dialogs as `<`*`instance_name`*`>_wfreq`, where `<`*`instance_name`*`>` is the schematic instance name of the *wlan11p* cell.

© 1994-2020
All Rights Reserved.

Frame structure

A `wlan11p` frame consists of preamble and data. The Data length property can be set with a default of 80 symbols. The frame duration and sampling frequencies are given in the table below assuming 80 data symbols.

Bandwidth	Preamble (time) (STF+LTF)	Signal	Data (time for 20 symbols)	Total time	fftlength	TimeStep
10M	32u	8u	640u	Preamble+ Signal+ Data	Oversample* 64	1/10M/Oversample
5M	64u	16u	1280u	Preamble+ Signal+ Data	Oversample* 64	1/5M/Oversample

Bandwidth	Constellation	Data Carriers (of 1 symbol)	Pilots (of 1 symbol)
All	Up to 64-QAM	48	4

© 1994-2020 All Rights Reserved.

CDF Parameters

Label	Parameter	Type	Default
Average Power (dbm)	dbm	Double	6.99
Resistance	r	Double	50
Number of Frames	framenumber	Double	1
Oversample ratio	oversample	Integer	4
SNR (db)	snr	Double	N/A (Infinite)
Band	WLAN11P_Band	Enum	5G
Bandwidth	WLAN11P_Bandwidth	Enum	10M
Channel Number	WLAN11P_Channel	Integer	171
PN Code	WLAN11P_DataType	Enum	PN9
Number of Data Symbols	WLAN11P_DataSym	Integer	80
Modulation Coding Scheme	WLAN11P__MCS	Integer	7
EVM measurement type	WLAN11P_EVMType	Enum	Data&Pilot

Parameter Descriptions

- *Average Power*: Source available average power in dBm.
- *Resistance*: Source Thevenin resistance in Ohms.
- *Number of frames*: The number of frames to simulate.
- Oversample: Controls the sampling frequency. The default is 4.
- *SNR*: Signal-to-noise power ratio.
- *Band*: Frequency band. Possible values are 5G or 700M.
- *Bandwidth*: Channel Bandwidth for the 5G band. Can be set to 5M or 10M. For 20M bandwidth, use the wlan11a source.

© 1994-2020 All Rights Reserved.

- *Channel number*: Channel number in range [170:184] in the 5GHz band. For the 10MHz channel bandwidth, the carrier frequency is given by `Fc (MHz) = 5000 + 5*WLAN11P_Channel`. For the 5MHz channel bandwidth, the carrier frequency is given by `Fc (MHz) = 5002.5 + 5*WLAN11P_Channel`.
- *PN Code*: Pseudo-random code shift register length. Default is PN9. PN15 and PN23 are also available.
- *Number of data symbols*: Default is 80.
- *Modulation Coding Scheme*: Default is 7. MCS 0 and 1 are bpsk. MCS 2 and 3 are qpsk. MCS 4 and 5 are 16-QAM. MCS 6 and 7 are 64-QAM. Note that coding is not supported. Coding is used for error detection and correction. All the bits are treated as data bits.
- *EVM measurement type*: When set to `Data&Pilot`, the EVM calculation includes all the used subcarriers of Data and Pilots. When set to `Data`, the EVM calculation includes only the Data subcarriers. When set to `Pilot`, the EVM calculation includes only the Pilot subcarriers. Possible values are `Data&Pilot`, `Data`, and `Pilot`.

In the ADE Direct Plot functions, the spectral mask follows 802.11p-2010 Table I.7. Power Class C is adopted.

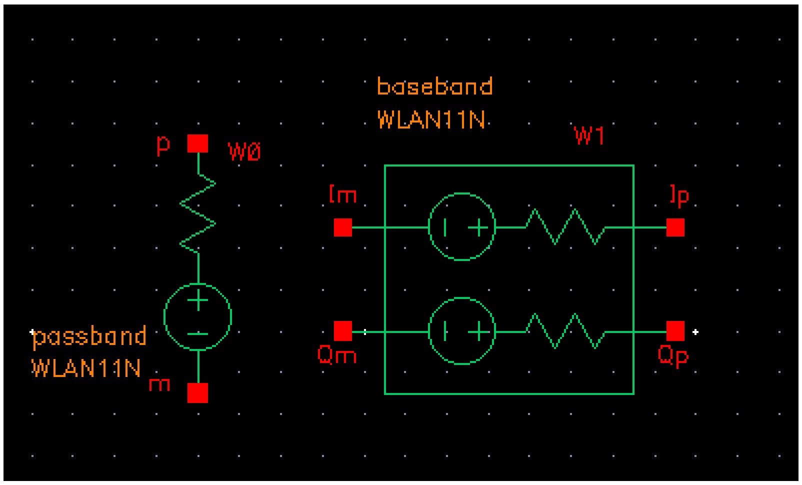
Example

```
W0 (net2 net1) wsource standard="WLAN11P" r=50 dbm=6.99 WLAN11P_Bandwidth=10M
WLAN11P_Oversample=4 framenumber=1 WLAN_11P_Channel=174 WLAN_11PDataSym=80
```


© 1994-2020 All Rights Reserved.

WLAN11N

SYMBOL



802.11n modulation source

Command-line help

```
spectre -h wsource
```

Notes

`wlan11n` applies 802.11n-compliant modulation according to *IEEE Standard 802.11n-2009, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, Amendment 5: Enhancements for Higher Throughput*.

`wlan11n` netlists as a Spectre `wsource`.

Frequency Band

The carrier frequency is selectable to be in the 2.4GHz or 5 GHz band. As always, you can reference the carrier frequency in the schematic and analysis dialogs as <*instance_name*>`_wfreq`, where <*instance_name*> is the schematic instance name of the *wlan11n* cell.

Frame structure

© 1994-2020 All Rights Reserved.

A `wlan11n` frame consists of preamble and data. Data length is settable with a property on the schematic symbol. The frame duration and sampling frequencies are given in the table below.

<table>
<tr><th></th><th>Preamble (time)</th><th>Data (time for 10 symbols)</th><th>Total time</th><th>fftlength</th><th>Time Step</th></tr>
<tr><td>20M Mix</td><td rowspan="2">36u</td><td rowspan="6">40u for Long GI
36u for Short GI</td><td rowspan="6">Preamble +Data</td><td>Oversample*64</td><td>1/20M/Oversample</td></tr>
<tr><td>40M Mix</td><td>Oversample*128</td><td>1/40M/Oversample</td></tr>
<tr><td>20M Legacy</td><td rowspan="2">20u</td><td>Oversample*64</td><td>1/20M/Oversample</td></tr>
<tr><td>40M Legacy</td><td>Oversample*128</td><td>1/40M/Oversample</td></tr>
<tr><td>20M Green</td><td rowspan="2">28u</td><td>Oversample*64</td><td>1/20M/Oversample</td></tr>
<tr><td>40M Green</td><td>Oversample*128</td><td>1/40M/Oversample</td></tr>
</table>

© 1994-2020 All Rights Reserved.

	Constellation	Data Carriers (of 1 symbol)	Pilots (of 1 symbol)
20M Mix		52	4
40M Mix		108	6
20M Legacy		48	4
40M Legacy		100	6
20M Green		52	4
40M Green	64QAM	108	6

© 1994-2020 All Rights Reserved.

CDF Parameters

Label	**Parameter**	**Type**	**Default**
Band Type	N/A	Enum	Baseband
Power (dbm)	dbm	Double	6.99
Resistance	r	Double	50
Number of frames	framenumber	Integer	1
Oversample ratio	oversample	Integer	4
SNR (db)	snr	Double	N/A (Infinite)
PHY Format	WLAN1N_Mode	Enum	Mix
Guard Interval	WLAN11N_GI	Enum	Long
Bandwidth	WLAN11N_Bandwidth	Enum	20M
Frequency Band	WLAN11N_Band	Enum	5G
Channel Number	WLAN11N_Channel	Integer	1
Number of Data Symbols	WLAN11N_DataSym	Integer	20
EVM	WLAN11N_EVMType	Enum	Data&Pilot

Parameter Descriptions

- *Band type*: Sets the passband or baseband operating mode.
- *Average Power*: Source available average power in dBm.
- *Resistance*: source Thevenin resistance in Ohms.
- *Number of frames*: The number of frames to simulate. Refer to the table for the frame structure and details.
- *PHY format*: Choice of *Mix*, *Green*, or *Legacy*.
- *Bandwidth*: Choice of 20 MHz or 40 MHz bandwidth.
- *Frequency band:* Choice of 2.4G or 5G band.
- *Oversample*: Controls the sampling frequency. Refer to the table for details.
- *Guard interval*: Choice of Long and Short guard interval.

© 1994-2020 All Rights Reserved.

- *SNR*: Signal-to-noise power ratio.
- *Channel number*: Channel number in range [0:200]. For the 5GHz band, the carrier frequency is given by `Fc (MHz) = 5000 + 5*WLAN11N_Channel`. For the 2.4GHz band, it is given by `Fc (MHz) = 2407+ 5*WLAN11N_Channel`.
- *EVM measurement type*: EVM measurement type. When set to `Data&Pilot`, the EVM calculation includes all the used subcarriers of Data and Pilots. When set to `Data`, the EVM calculation includes only the Data subcarriers. When set to `Pilot`, the EVM calculation includes only the Pilot subcarriers. Possible values are `Data&Pilot`, `Data`, and `Pilot`.

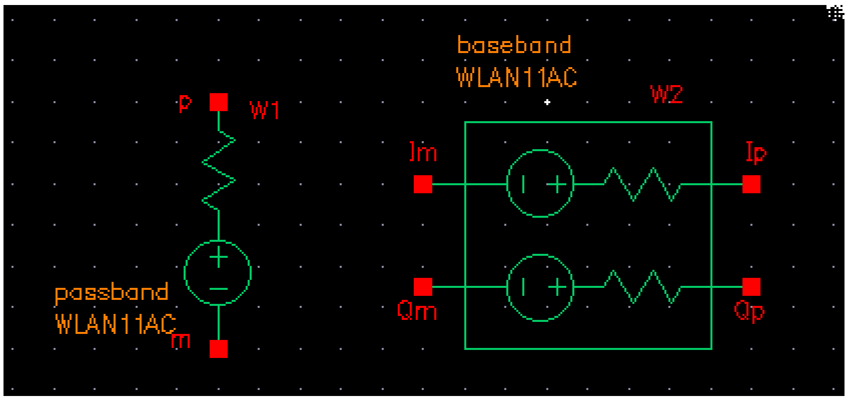
Example

```
W0 (net2 net1) wsource standard="WLAN11N" r=50 dbm=6.99 WLAN11N_Mode=Mix \
WLAN11N_Bandwidth=20M oversample=4 WLAN11N_GI=Long framenumber=1
```


© 1994-2020 All Rights Reserved.

WLAN11AC

SYMBOL



802.11ac modulation source

Command-line help

```
spectre -h wsource
```

Notes

`wlan11ac` applies 802.11ac-compliant modulation according to *IEEE Standard 802.11ac-2013, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, Amendment 4: Enhancements for Very High Throughput for Operation in Bands below 6GHz*.

`wlan11ac` netlists as a Spectre `wsource`.

Frequency Band

The carrier frequency is in the 5 GHz band. As always, you can reference the carrier frequency in the schematic and analysis dialogs as `<`*`instance_name`*`>_wfreq`, where `<`*`instance_name`*`>` is the schematic instance name of the *wlan11ac* cell.

© 1994-2020 All Rights Reserved.

Frame structure

A `wlan11ac` frame consists of preamble and data. Data length is settable with a property with a default of 80 symbols. The frame duration and sampling frequencies are given in the table below assuming 80 symbols.

	Preamble (time)	Data (time for 80 symbols)	Total time	fftlength	TimeStep
20M	40u	320u for Long GI 288u for Short GI	Preamble+ Data	Oversample* 64	1/20M/Oversa mple
40M	40u			Oversample* 128	1/40M/Oversa mple
80M	40u			Oversample* 256	1/80M/Oversa mple
160M	40u			Oversample* 512	1/160M/Overs ample
80M+80M	40u			Oversample* 256	1/80M/Oversa mple

© 1994-2020 All Rights Reserved.

	Constellation	Data Carriers (of 1 symbol)	Pilots (of 1 symbol)
20M		52	4
40M		108	6
80M		234	8
160M		468	16
80M+80M	BPSK, QPSK, 16-QAM, 64QAM, 256-QAM	468	16

© 1994-2020
All Rights Reserved.

CDF Parameters

Label	**Parameter**	**Type**	**Default**
Band Type	N/A	Enum	Baseband
Average Power (dbm)	dbm	Double	6.99
Resistance	r	Double	50
Number of Frames	framenumber	Double	1
Oversample ratio	oversample	Integer	4
SNR (db)	snr	Double	N/A (Infinite)
Guard Interval	WLAN11AC_GI	Enum	Long
Bandwidth	WLAN11AC_Bandwidth	Double	80M
Channel Number	WLAN11AC_Channel	Integer	0
PN Code	WLAN11AC_DataType	Enum	PN23
Number of Data Symbols	WLAN11AC_DataSym	Integer	80
Modulation Coding Scheme	WLAN11AC_MCS	Integer	8
EVM	WLAN11AC_EVMType	Enum	Data&Pilot

Parameter Descriptions

- *Band type*: Sets the passband or baseband operating mode.
- *Average Power*: Source available average power in dBm.
- *Resistance*: Source Thevenin resistance in Ohms.
- *Number of frames*: The number of frames to simulate. Refer to the table for the frame structure and details.
- *Bandwidth*: Choice of 20M, 40M, 80M, 160M, or 80+80M bandwidth.
- *Modulation coding scheme:* 0 through 9 are selectable.
- *Oversample*: Controls the sampling frequency. Refer to the table for details.
- *Guard interval*: Choice of Long and Short guard interval.

© 1994-2020 All Rights Reserved.

- *SNR*: Signal-to-noise power ratio.
- *Channel number*: Channel number in range [0:200]. The carrier frequency is given by `Fc (MHz) = 5000 + 5*WLAN11N_Channel`.
- PN code: This sets the length of the shift register in the pseudorandom number generator to generate the data bits. The default is PN23 and is strongly suggested. PN23, PN15, and PN9 are available.
- *EVM measurement type*: EVM measurement type. When set to `Data&Pilot`, the EVM calculation includes all the used subcarriers of Data and Pilots. When set to `Data`, the EVM calculation includes only the Data subcarriers. When set to `Pilot`, the EVM calculation includes only the Pilot subcarriers. Possible values are `Data&Pilot`, `Data`, and `Pilot`.

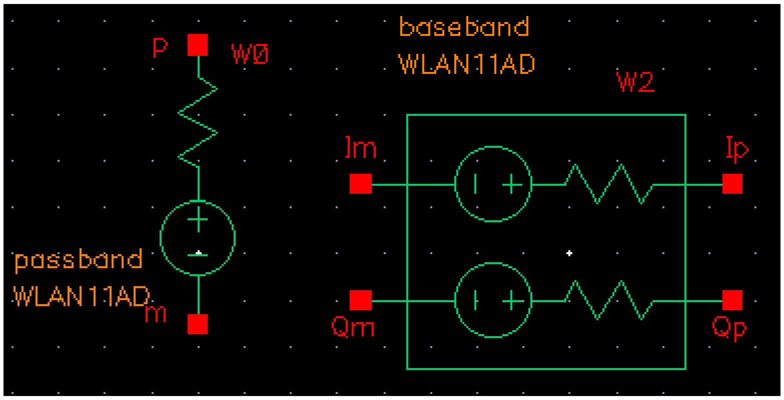
Example

```
W0 (net2 net1) wsource standard="WLAN11AC" r=50 dbm=6.99 WLAN11AC_MCS=8 \
WLAN11AC_Bandwidth=80M oversample=4 WLAN11AC_GI=Long framenumber=1
```


© 1994-2020 All Rights Reserved.

WLAN11AD

SYMBOL



802.11ad modulation source

Command-line help

```
spectre -h wsource
```

Notes

`wlan11ad` applies 802.11ad-compliant modulation according to *IEEE Standard 802.11ad-2012, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, Amendment 3: Enhancements for Very High Throughput in the 60GHz band*.

`wlan11ad` netlists as a Spectre `wsource`.

Frequency Band

The carrier frequency is in the 60 GHz band range. As always, you can reference the carrier frequency in the schematic and analysis dialogs as *<instance_name>*`_wfreq`, where *<instance_name>* is the schematic instance name of the *wlan11ad* cell.

© 1994-2020
All Rights Reserved.

Root-Raised Cosine Filter

A root-raised cosine (RRC) filter with roll-off factor in the range (0:1) can be chosen. Default is 0.25. The impulse response of RRC filter is given by:

$$h(t) = 4\alpha\frac{\cos[(1+\alpha)\pi t/T] + \dfrac{\sin[(1-\alpha)\pi t/T]}{4\pi\alpha t/T}}{\pi\sqrt{T}[1-(4\alpha t/T)^2]}$$

where T is the symbol period and α is the roll-off factor.

The length of impulse response of the filter is given by:

$$n = span \times oversample + 1$$

where *span* is the filter span in symbols and *oversample* is the output samples per symbol.

© 1994-2020 All Rights Reserved.

Frame structure

A wlan11ad frame consists of preamble, header, and data. Details are provided in the table below.

Layer	Preamble	Header	Data	Data Modulation	FrameLength (FL)	FrameTime	Sampling Rate
Control PHY (MCS [0])	7552 symbols	8192 pi/2 DBPSK symbols	[14:1023] octets Ga(32) spreading Pi/2 rotation	pi/2 DBPSK	Without RRC: FL1= 7552+8192+DataLength*8*32 With RRC: FL=FL1+20 Here 20 is the span of RRC filter	FrameLength/Sampling Rate	1760M
SC PHY (MCS in [1:12] range)	3328 pi/2 BPSK symbols	1024 pi/2 BPSK symbols	[1:262143] octets	pi/2 DBPSK pi/2 QPSK pi/2 16 QAM	Without RRC: FL1= 3328+1024+DataLength *8/Bitspersymbol+ Guard bits Bitspersymbol is {1,2,4} corresponding to {BPSK, QPSK, 16 QAM} Data blocking: Guard interval=64 symbols. Data=448 symbols. With RRC: FL=FL1+20 Here, 20 is the span of RRC filter	FrameLength/Sampling Rate	1760M
OFDM PHY (MCS in [13:24] range)	3328 Pi/2 BPSK symbols upsampled from 1.76GHz to 2.64 GHz and filtered.	1 QPSK OFDM symbol	N_{SYM}	SQPSK, QPSK, 16 QAM, 64 QAM	FL=FrameTime*SamplingRate	(2.133+0.242* N_{SYM}) µs	2640M

© 1994-2020 All Rights Reserved.

CDF Parameters

Label	Parameter	Type	Default
Band Type	N/A	Enum	passband
Average Power (dbm)	dbm	Double	6.99
Resistance	r	Double	50
Number of Frames	framenumber	Double	1
Oversample ratio	oversample	Integer	4
SNR (db)	snr	Double	N/A (Infinite)
Channel Number	WLAN11AD_Channel	Integer	1
Modulation Coding Scheme	WLAN11AD_MCS	Integer	12
Number of data symbols	WLAN11AD_DataSym	Integer	80
EVM measurement type	WLAN11AD_EVMType	Enum	Data&Pilot
Data length in octets	WLAN11AD_DataLength	Integer	100
Pulse shaping filter type	WLAN11AD_FilterType	Enum	RRC
Roll-off factor	RRC Rolloff Factor	Double	0.25

Parameter Descriptions

- *Band type*: Sets the passband or baseband operating mode.
- *Average Power*: Source available average power in dBm.
- *Resistance*: Source Thevenin resistance in Ohms.
- *Number of frames*: The number of frames to simulate. Refer to the table for the frame structure and details.
- *Oversample ratio*: Controls the sampling frequency.
- *SNR*: Signal-to-noise power ratio.

© 1994-2020 All Rights Reserved.

- *Channel number*: Channel number in range [1:4]. The carrier frequency is given by `Fc (GHz) = 56.16 + 2.16*WLAN11N_Channel`.
- *Modulation and coding scheme*: 0 through 24 are selectable.
- Number of data symbols: The number of data symbols. Default is 80. This option is visible only when MCS is in the range [13:24].
- EVM measurement type: EVM measurement type. When set to `Data&Pilot`, EVM calculation includes all the used subcarriers of Data and Pilots. When set to `Data`, EVM calculation includes only the Data subcarriers. When set to `Pilot`, EVM calculation includes only the Pilot subcarriers. Possible values are `Data&Pilot`, `Data`, and `Pilot`. This option is visible only when MCS is in the range [13:24].
- *Data length in octets*: Sets the length of the data portion of the frame in octets. This option is visible only when MCS in the range [0:12]
- *Pulse shaping filter type*: Sets the pulse shaping filter type. Possible values are *RRC* and *none*. This option is visible only when MCS is in the range [0:12].
- Roll-off factor: Roll-off factor for the RRC filter.

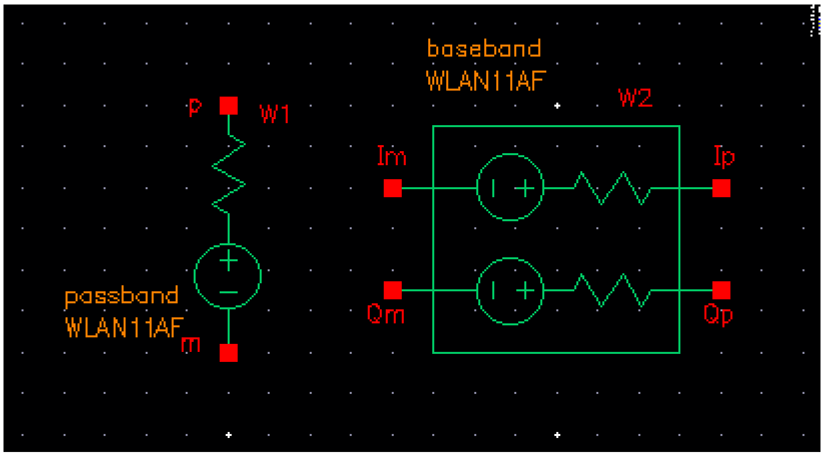
Example

```
W0 (net2 net1) wsource standard="WLAN11AD" r=50 dbm=6.99 WLAN11AD_MCS=8 \
oversample=4 WLAN11AD_FilterType=RRC
```


© 1994-2020 All Rights Reserved.

WLAN11AF

SYMBOL



802.11af modulation source

Command-line help

```
spectre -h wsource
```

Notes

`wlan11af` applies 802.11af-compliant modulation according to *IEEE Standard 802.11af-2013, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, Amendment 5: TV White Spaces Operation*.

`wlan11af` netlists as a Spectre `wsource`.

Frequency Band

The carrier frequency is in the television white space frequency range. As always, you can reference the carrier frequency in the schematic and analysis dialogs as <*instance_name*>`_wfreq`, where <*instance_name*> is the schematic instance name of the *wlan11af* cell.

© 1994-2020 All Rights Reserved.

Frame structure

A `wlan11af` frame consists of preamble and data. Data length is settable with a property with a default of 80 symbols. The frame duration and sampling frequencies are given in the table below assuming 80 symbols.

	Preamble (time)	Data (time for 80 symbols)	Total time	fftlength	TimeStep
6M	300u	2.4m for Long GI 2.16m for Short GI	Preamble+Data	Oversample* 144	1/6M/Oversample
7M	300u	2.4m for Long GI 2.16m for Short GI		Oversample* 168	1/7M/Oversample
8M	225u	1.8m for Long GI 1.62m for Short GI		Oversample* 144	1/8M/Oversample

© 1994-2020 All Rights Reserved.

	Constellation	Data Carriers (of 1 symbol)	Pilots (of 1 symbol)
6M	BPSK, QPSK, 16-QAM, 64QAM, 256-QAM	108	6
7M		108	6
8M		108	6

CDF Parameters

Label	**Parameter**	**Type**	**Default**
Band Type	N/A	Enum	Baseband
Average Power (dbm)	dbm	Double	6.99
Resistance	r	Double	50
Number of frames	framenumber	Integer	1
Oversample ratio	oversample	Integer	4
SNR (db)	snr	Double	N/A (Infinite)
Guard Interval	WLAN11AF _GI	Enum	Long
Bandwidth	WLAN11AF _Bandwidth	Enum	6M
Channel Number	WLAN11AF _Channel	Integer	2

© 1994-2020 All Rights Reserved.

PN Code	WLAN11AF_DataType	Enum	PN23
Number of Data Symbols	WLAN11AF_DataSym	Integer	80
Modulation Coding Scheme	WLAN11AF_MCS	Integer	8
EVM	WLAN11AF_EVMType	Enum	Data&Pilot
Carrier freq for 7M/ 8M BW	WLAN11AF_Frequency	Enum	233.5M

Parameter Descriptions

- *Band type*: Sets the passband or baseband operating mode.
- *Average Power*: Source available average power in dBm.
- *Resistance*: Source Thevenin resistance in Ohms.
- *Number of frames*: The number of frames to simulate. Refer to the table for the frame structure and details.
- *Bandwidth*: Choice of 6 MHz, 7 MHz, or 8 MHz bandwidth.
- *Modulation coding scheme:* 0 through 9 are selectable.
- *Oversample*: Controls the sampling frequency. Refer to the table for details.
- *Guard interval*: Choice of Long and Short guard interval.
- *SNR*: Signal-to-noise power ratio.
- *Channel number*: Channel number in range [2:51]. The carrier frequency is given by `StartingFreq + ChNum*6M` where the StartingFreq is 45MHz for channels 2 through 4, 49MHz for channels 5 and 6, 99MHz for channels 7 through 13, and 389MHz for channels 14 through 51.
- PN code: This sets the length of the shift register in the pseudorandom number generator to generate the data bits. The default is PN23 and is strongly suggested. PN23, PN15, and PN9 are available.
- *EVM measurement type*: EVM measurement type. When set to `Data&Pilot`, the EVM calculation includes all the used subcarriers of Data and Pilots. When set to `Data`, the EVM calculation includes only the Data subcarriers. When set to `Pilot`, the EVM calculation includes only the Pilot subcarriers. Possible values are `Data&Pilot`, `Data`, and `Pilot`.

© 1994-2020 All Rights Reserved.

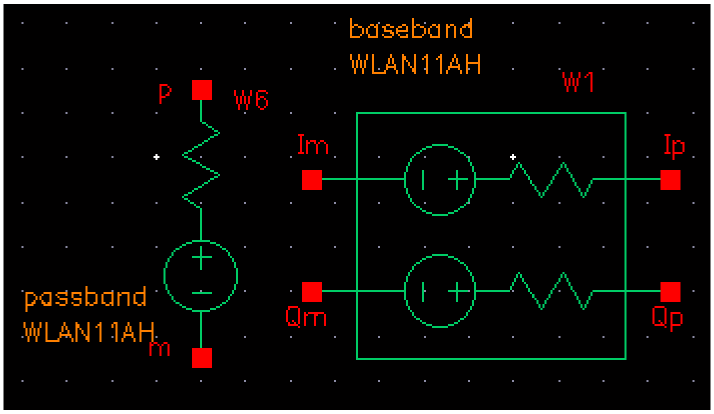
- *Carrier freq for 7M/8M BW*: When the bandwidth is 7MHz or 8MHz, the carrier frequency must be specified.

Example

```
W0 (net2 net1) wsource standard="WLAN11AF" r=50 dbm=6.99 WLAN11AF_MCS=8 \
WLAN11AF_Bandwidth=6M oversample=4 WLAN11AF_GI=Long framenumber=1
```


© 1994-2020 All Rights Reserved.

WLAN11AH



802.11ah modulation source

Command-line help

`spectre -h wsource`

Notes

`wlan11ah` applies 802.11ah-compliant modulation that is an amendment of the IEEE 802.11-2007 wireless networking standard.

`wlan11ah` netlists as a Spectre `wsource`.

Frequency Band

The carrier frequency is set in the *Carrier frequency* field in the property list. An ADE variable `W0_wfreq` (where `W0` is the instance name of the wireless source) is introduced with a default value of `1G` in the ADE variables section. The value set in ADE is not used. Instead, this variable is assigned at the beginning of the simulation to the value specified in the *Carrier frequency* field.

Frame Structure

The WLAN11AH standard offers three modes: 1 MHz (S1G_1M), long preamble mode (S1G_LONG), and short preamble mode (S1G_SHORT).

© 1994-2020
All Rights Reserved.

A `wlan11ah` frame consists of preamble and data. The frame duration and sampling frequencies are given in the table below assuming 80 symbols.

	Preamble (time)	Data (time for 1 symbol)	fftlength	TimeStep
1M	600u	36u short GI 40u long GI	Oversample*32	1/1M/Oversample
2M	280u for short preamble type 360u for long preamble type		Oversample*64	1/2M/Oversample
4M			Oversample*128	1/4M/Oversample
8M			Oversample*256	1/8M/Oversample
16M			Oversample*512	1/16M/Oversample

	Constellation	Data Carriers (of 1 symbol)	Pilots (of 1 symbol)
1M	BPSK, QPSK, 16-QAM, 64-QAM, 256-QAM	24	2
2M		42	4
4M		108	6
8M		234	8
16M		468	16

© 1994-2020
All Rights Reserved.

CDF Parameters

Label	**Parameter**	**Type**	**Default**
Band Type	N/A	Enum	Baseband
Average Power (dbm)	dbm	Double	6.99
Resistance	r	Double	50
Number of frames	framenumber	Integer	1
Oversample ratio	oversample	Integer	4
SNR (db)	snr	Double	N/A (Infinite)
Carrier Frequency	WLAN11AH_Frequency	Enum	902M
PN Code	WLAN11AH_DataType	Enum	PN23
Number of data symbols	WLAN11AH_DataSym	Integer	80
Bandwidth	WLAN11AH _Bandwidth	Enum	8M
Modulation and coding scheme	WLAN11AH_MCS	Integer	8
Guard Interval	WLAN11AH _GI	Enum	Normal
EVM measurement type	WLAN11AH_EVMType	Enum	Data&Pilot
Preamble type (for 2M, 4M, 8M, 16M rate)	WLAN11AH_Preamble	Enum	Short

Parameter Descriptions

- *Band type*: Sets the passband or baseband operating mode.
- *Average Power*: Source available average power in dBm.
- *Resistance*: Source Thevenin resistance in Ohms.
- *Number of frames*: The number of frames to simulate. Refer to the table for the frame structure and details.
- *Oversample ratio*: Controls the sampling frequency. Refer to the table for details.

© 1994-2020
All Rights Reserved.

- *SNR*: Signal-to-noise power ratio.
- *Carrier Frequency*: Channel center frequency in the range (700M: 1G).
- PN code: This sets the length of the shift register in the pseudorandom number generator to generate the data bits. The default is PN23 and is strongly suggested. PN23, PN15, and PN9 are available.
- *Number of data symbols*: Number of PSDU data symbols in each frame.
- *Bandwidth*: Choice of 1MHz, 2 MHz, 4 MHz, 8 MHz, or 16MHz bandwidth.
- *Modulation coding scheme:* 0 through 9 are selectable.
- *Guard interval*: Choice of *Normal* and *Short* guard intervals.
- *EVM measurement type*: EVM measurement type. When set to `Data&Pilot`, the EVM calculation includes all the used subcarriers of Data and Pilots. When set to `Data`, the EVM calculation includes only the Data subcarriers. When set to `Pilot`, the EVM calculation includes only the Pilot subcarriers. Possible values are `Data&Pilot`, `Data`, and `Pilot`.
- *Preamble type*: Sets the preamble type *Short* or *Long* for the 2 MHz, 4 MHz, 8 MHz, or 16 MHz bandwidth.

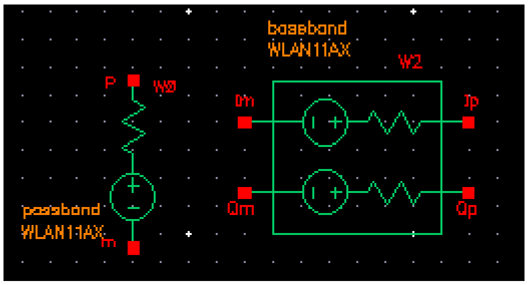
Example

```
W0 (net2 net1) wsource standard="WLAN11AH" r=50 dbm=6.99 WLAN11AH_MCS=8 \
WLAN11AH_Bandwidth=8M oversample=4 framenumber=1
```


© 1994-2020
All Rights Reserved.

WLAN11AX

Symbol



802.11ax modulation source

Command-line help

```
spectre -h wsource
```

Notes

`wlan11ax` netlists as a Spectre `wsource`.

Frequency Band

The 2.4GHz, 5GHz, and 6GHz ISM bands are used.

Frame Structure

A wlan11ax frame consists of preamble and data. Data length is settable with a property with a default of 80 symbols. The frame duration and sampling frequencies are given in the table below.

© 1994-2020 All Rights Reserved.

	Preamble (time)	Data time (Per Symbol) Excludes guard interval	FFT Length	Time Step	Total Time 80 symbols, G1	Total time 80 symbols GI4
20M	44u	12.8u	Oversample*64	1/20M/Oversample	1.132msec	1.324msec
40m	44u	12.8u	Oversample*128	1/40M/Oversample		
80M	44u	12.8u	Oversample*256	1/80M/Oversample		
160M	44u	12.8u	Oversample*512	1/160M/Oversample		
80M+80M	44u	12.8u	Oversample*256	1/80M/Oversample		

	Constellation	Data Carriers of 1 Symbol	Pilot Carriers
20M	BPSK, QPSK, 16-QAM, 64-QAM, 256-QAM, 1024-QAM	242	8
40m		484	16
80M		996	16
160M		2x996	2x16
80M+80M		2x996	2x16

© 1994-2020 All Rights Reserved.

CDF Parameters

Label	**Parameter**	**Type**	**Default**
Band Type	N/A	Enum	Baseband
Average Power (dbm)	dbm	Double	6.99
Resistance	r	Double	50
Number of Frames	framenumber	Double	1
Oversample ratio	oversample	Integer	4
SNR (db)	snr	Double	N/A (Infinite)
Frequency Band	WLAN11AX_Band	Enum	5G
Channel Number	WLAN11AX_Channel	Integer	0
PN Code	WLAN11AX_DataType	Enum	PN23
Number of Data Symbols	WLAN11AX_DataSym	Integer	80
Bandwidth	WLAN11AX_Bandwidth	Enum	80M
Modulation Coding Scheme	WLAN11AX_MCS	Integer	8
Guard Interval	WLAN11AX_GI	Enum	GI1
EVM measurement type	WLAN11AX_EVMType	Enum	Data&Pilot

Parameter Descriptions

- *Band type*: Sets the passband or baseband operating mode.
- *Average Power*: Source available average power in dBm.
- *Resistance*: Source Thevenin resistance in Ohms.
- *Number of frames*: The number of frames to simulate. Refer to the table for the frame structure and details.
- *Oversample ratio*: Controls the sampling frequency. Refer to the table for details.
- *SNR*: Signal-to-noise power ratio.

© 1994-2020 All Rights Reserved.

- Frequency Band: Choice of 2.4GHz, 5GHz, and 6GHz ISM bands.
- *Channel number*: Channel number. For 5G band, the carrier frequency is given by `Fc (MHz) = 5000 + 5*WLAN11N_Channel`. For 2.4G band, the carrier frequency is given by `Fc (MHz) = 2407 + 5*WLAN11N_Channel`. For 6G band, it is given by `Fc = 5925 + 5*WLAN11N_Channel.`
- PN code: This sets the length of the shift register in the pseudorandom number generator to generate the data bits. The default is PN23 and is strongly suggested. PN23, PN15, and PN9 are available.
- *Bandwidth*: Signal bandwidth. For 5G and 6G band, choice of 20M, 40M, 80M, 160M, or 80+80M bandwidth is available. For 2.4G band, 20M, 40M, or 80M bandwidth is available.
- *Modulation coding scheme:* 0 through 11 are selectable.
- *Guard interval*: Choice of GI1, GI2, and GI4 guard intervals. GI1 is 0.8usec. GI2 is 1.6usec. GI4 is 3.2usec.
- *EVM measurement type*: EVM measurement type. When set to `Data&Pilot`, the EVM calculation includes all the used subcarriers of Data and Pilots. When set to `Data`, the EVM calculation includes only the Data subcarriers. When set to `Pilot`, the EVM calculation includes only the Pilot subcarriers. Possible values are `Data&Pilot`, `Data`, and `Pilot`.

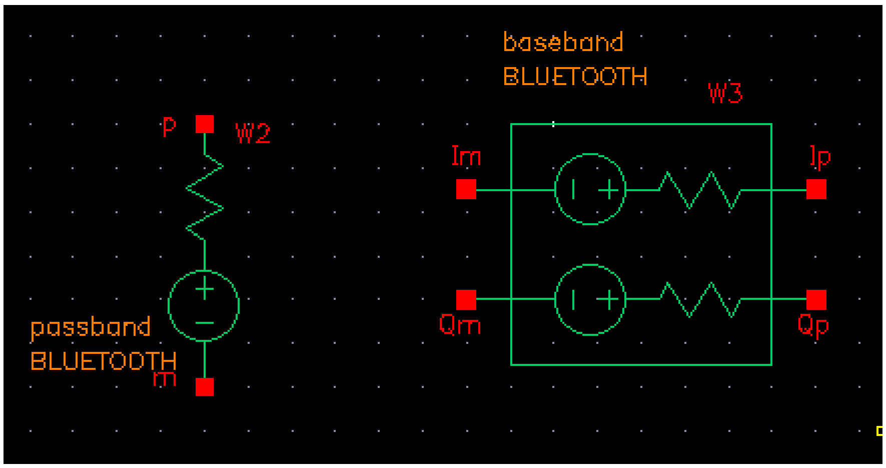
Example

```
W0 (net2 net1) wsource standard="WLAN11AX" r=50 dbm=6.99 WLAN11AX_MCS=8 \
WLAN11AX_Bandwidth=80M oversample=4 WLAN11AX_GI=GI1 framenumber=1
```


© 1994-2020 All Rights Reserved.

Bluetooth

SYMBOL



Bluetooth modulation source

Command-line help

`spectre -h wsource`

Notes

`bluetooth` applies Bluetooth LE and Bluetooth HS compliant modulation according to *Covered Core Package Version 4.2, December 2014.*

`Bluetooth` netlists as a Spectre `wsource`.

Frequency Band

The 2.4GHz ISM band is used.

© 1994-2020
All Rights Reserved.

Frame structure

LE Mode

A `Bluetooth LE` frame consists of preamble, data, and CRC fields. Data length can be set using a property with a default of 10 octets. The frame duration is given in the table below assuming 10 octets.

Data Rate	Preamble (time)	Data (time for 10 octets)	CRC	Total time	Modulation	TimeStep
1M	40u	80u	24u	Preamble+Data+CRC	GFSK	1/2M/Oversample

HS Mode

A `Bluetooth HS` frame consists of preamble, header, and data. Data length can be set using a property with a default of 10 octets for data rates of 1M, 2M, 5.5M, and 11M. For 72M,

© 1994-2020
All Rights Reserved.

the default is 80 OFDM symbols. The frame duration is given in the table below assuming 10 octets for the 1M, 2M, 5.5M, and 11M data rates, and 80 OFDM symbols for the 72M rate.

Data Rate	Preamble+Header (time)	Data (time for 10 octets)	Total time	Modulation	TimeStep
1M	192u	80u	Preamble+Header+Data	DBPSK 11 Chips/Symbol	1/11M/Oversample
2M	192u	40u	Preamble+Header+Data	DQPSK 11 Chips/Symbol	1/11M/Oversample
5.5M	192u	14.54u	Preamble+Header+Data	CCK 8 Chips/Symbol	1/11M/Oversample
11M	192u	7.27u	Preamble+Header+Data	CCK 8 Chips/Symbol	1/11M/Oversample
72M	20u	320u for 80 OFDM symbols	Preamble+Header+Data	OFDM QAM-64	1/20M/Oversample

CDF Parameters

Label	**Parameter**	**Type**	**Default**
Band Type	N/A	Enum	Baseband

© 1994-2020 All Rights Reserved.

Average Power (dbm)	dbm	Double	6.99
Resistance	r	Double	50
Number of frames	framenumber	Integer	1
Oversample ratio	oversample	Integer	4
SNR (db)	snr	Double	N/A (Infinite)
Bluetooth Mode	BLUETOOTH_Mode	Enum	LE
Channel Number	BLUETOOTH _Channel	Integer	1
PN Code	BLUETOOTH_DataType	Enum	PN9
Data length in octets (LE Mode and HS Mode for 11M rate and below.)	BLUETOOTH_DataLength	Integer	10
Modulation type	BLUETOOTH_Modulation	Enum	GFSK
Modulation index	BLUETOOTH_MT	Double	0.5
Data rate for HS mode in bps	BLUETOOTH_DataRate	Integer	72M
Pulse shaping filter type (HS mode for 11M rate and below)	BLUETOOTH_FilterType	Enum	Gaussian
Bandwidth time product (HS mode for 11M rate and below)	BLUETOOTH_BT	Double	0.5
Number of data symbols (HS Mode, 72M data rate)	BLUETOOTH_DataSym	Enum	80
EVM measurement type (HS Mode, 72M data rate)	BLUETOOTH_EVMType	Enum	Data&Pilot

© 1994-2020
All Rights Reserved.

Parameter Descriptions

- *Band type*: Sets the passband or baseband operating mode.
- *Average Power*: Source available average power in dBm.
- *Resistance*: Source Thevenin resistance in Ohms.
- *Number of frames*: The number of frames to simulate. Refer to the table for the frame.
- *Oversample*: Controls the sampling frequency. Refer to the table for details.
- *SNR*: Signal-to-noise power ratio.
- *Bluetooth mode:* Sets the mode to LE, HS, EDR, or BR.
- *Channel number*: Channel number. For LE mode, it is in the range [0:39] and the carrier frequency is given by `Fc (MHz) = 2402 + 2 * BLUETOOTH_Channel`. For HS mode 72Mbps, it is in the range [1:13] and carrier frequency is given by `Fc (MHz) = 2407 + 5 * BLUETOOTH_Channel`. For other data rate in HS mode, it is in the range [1:14] and the carrier frequency is given by `Fc (MHz) = 2407 + 5 * BLUETOOTH_Channel` for channel [1:13], and `Fc = 2484` MHz for channel 14. For EDR and BR mode, it is in the range [0:78] and the carrier frequency is given by `Fc (MHz) = 2402 + BLUETOOTH_Channel`.
- *PN code:* This sets the length of the shift register in the pseudorandom number generator to generate the data bits. The default is PN9. PN23, PN15, and PN9 are available.
- *Data length in octets:* Data length in octets. For LE data channel, it is in the range [1:255]. For LE advertising channel (channel 0, 12, 39), it is in the range [6:37]. For HS mode, it is in the range [2:2e4] for simulation. For EDR mode, it is in the range [1:1021]. For BR mode, it is in the range [1:339].
- *Modulation type*: Modulation type. Possible values are `FSK` and `GFSK`.
- *Modulation index*: Modulation index. For LE mode, it is in the range [0.45:0.55]. For BR mode, it is in the range [0.28:0.35].
- *Data rate in bps*: Data rate in bps for LE, HS and EDR modes. Possible values are 1M and 2M for LE mode; 1M, 2M, 5.5M, 11M and 72M for HS mode; 2M and 3M for EDR mode.
- *Pulse shaping filter type*: This sets the pulse shaping filter type for HS mode for 11M rate and below. Possible values are None and Gaussian.
- *Bandwidth product*: Sets the bandwidth time product for HS mode for 11M rage and below in the range (0 1].

© 1994-2020 All Rights Reserved.

- *Number of data symbols*: This sets the number of data symbols in the frame for HS 72M data rate mode.
- *EVM measurement type*: When set to `Data&Pilot`, the EVM calculation includes all the used subcarriers of Data and Pilots. When set to `Data`, the EVM calculation includes only the Data subcarriers. When set to `Pilot`, the EVM calculation includes only the Pilot subcarriers. Possible values are `Data&Pilot`, `Data`, and `Pilot`. This is available for HS mode 72M data rate only.

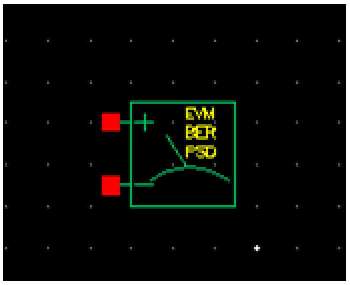
Example

```
W0 (net2 net1) wsource standard=”BLUETOOTH" r=50 dbm=6.99 \
BLUETOOTH_Mode=LE BLUETOOTH_Channel=2 BLUETOOTH_DataType=PN9 \
BLUETOOTH_DataLength=10 framenumber=1
```


© 1994-2020 All Rights Reserved.

WPROBE

SYMBOL



Wireless probe

Command-line help

`spectre -h wprobe`

Notes

`wprobe` serves the following two purposes:

1. In fast envelope wireless analysis mode, `wprobe` sets the output node.
2. In both regular and fast envelope mode, `wprobe` performs signal post processing to calculate EVM, BER, and Spectrum measurements, including ACPR.

Electrically, wprobes are open-circuited.

Measurement functions

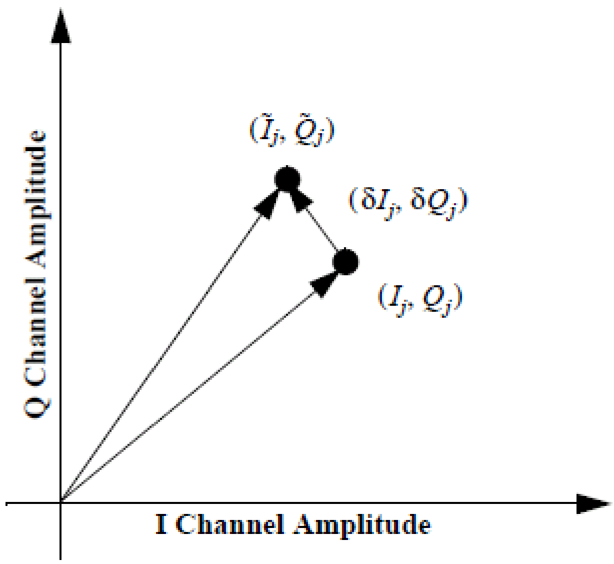
`wprobe` provides output for three functions for use in the *Direct Plot Form*: *Measure*, *Constellation*, and *Spectrum*.

Measure

Measure provides the EVM and BER measurements. They are shown in the *Direct Plot Form* as `<probe instance name>.evm` and `<probe instance name>.ber`. Both are computed and plotted as functions of the frame index.

© 1994-2020 All Rights Reserved.

EVM measures the modulation accuracy of the transmitter. In order to calculate EVM, a time record of N received complex samples $(\tilde{I}_j, \tilde{Q}_j)$ is captured. For each received complex sample, a decision is made about which complex value was transmitted. The error vector $\delta I_j, \delta Q_j$ is defined as the distance from the ideal position (I_j, Q_j) to the actual position of the received point.



Therefore, the received vector is the sum of the ideal vector and the error vector, as shown below.

$$(\tilde{I}_j, \tilde{Q}_j) = (I_j, Q_j) + (\delta I_j, \delta Q_j),$$

and EVM is defined as:

$$EVM = \sqrt{\frac{\frac{1}{N}\sum_{j=1}^{N}(\delta I_j^2 + \delta Q_j^2)}{S^2}} \text{x}100\%$$

S is the magnitude of the vector to the ideal constellation.

$$S^2 = \frac{1}{N}\sum_{j=1}^{N}(I_j^2 + Q_j^2)$$

In digital transmission, the number of bit errors is the number of received bits of a data stream over a communication channel that have been altered due to noise, interference, distortion, or bit synchronization errors.

© 1994-2020 All Rights Reserved.

The bit error rate or bit error ratio (BER) is the number of bit errors divided by the total number of transferred bits during the simulated time interval. BER is a unitless performance measure, often expressed as a percentage.

Constellation

The *Constellation* function outputs two waveforms, `<probe instance name>.mea` and *<probe instance name>*`.ref`. `<probe instance name>.mea` plots $(\tilde{I}_j, \tilde{Q}_j)$, and the measured I/Q samples across the probe. The `.ref` measurement is the ideal reference signal from `wsource`.

Spectrum

The *Spectrum* measurement calculates the power spectral density of the signal across the probe terminals using the `psdbb()` Calculator function. It also calculates the ACPR, Main Channel Power, and displays the spectral mask.

`psdbb()` function parameters, such as From, To, Number of Samples and Window Size are obtained automatically from the 'param' dataset.

The Window Size is the largest value of `pow(2, n)`, where `n` satisfies:

$$2^n < 2\frac{F_{max}}{F_{min}}$$

`Fmax = 1/Tstep` and `Fmin` is the frequency bin width. Taking `wlan11n` as an example, `Fmin` is equal to one subcarrier bandwidth / oversample ratio.

The Window type is fixed at Cosine4.

FSKErr

When the modulation is FSK, select `yes` for `FSKErr`. This will toggle off the EVM measurement as they are mutually exclusive. Setting this to `yes` allows plotting the deviation from center for the data part of the frame.

CDF Parameters

Label	Parameter	Type	Default

© 1994-2020 All Rights Reserved.

EVM	evm	String	yes
BER	ber	String	yes
PSD	psd	String	yes
FSKErr	fskerr	string	no specification
Additional Measure	N/A	String	

Example

```
WPRB0 (net2 net1) wprobe ber=yes psd=yes evm=yes
```


© 1994-2020 All Rights Reserved.

Modifying the BB Signal Generators Using Modelwriter

The baseband signal generators use FIR (finite impulse response) filters to shape their output pulses. Shaped output pulses serve several purposes:

- They help keep the transmitted signal inside the specified band.
- They work with their receiver counterparts to maximize the signal-to-noise ratio.
- Together with their receiver counterparts, they satisfy the Nyquist sampling criterion for an intersymbol-interference-free channel.

The Modelwriter gives you a convenient user-interface for creating variations on the baseband signal generators in the library. The most likely variation is in the FIR filter. This section explains how to use the Modelwriter to create a new GSM generator with different FIR filters.

1. Bring up the Modelwriter and do the following:

    a. Double click the Telecom folder.

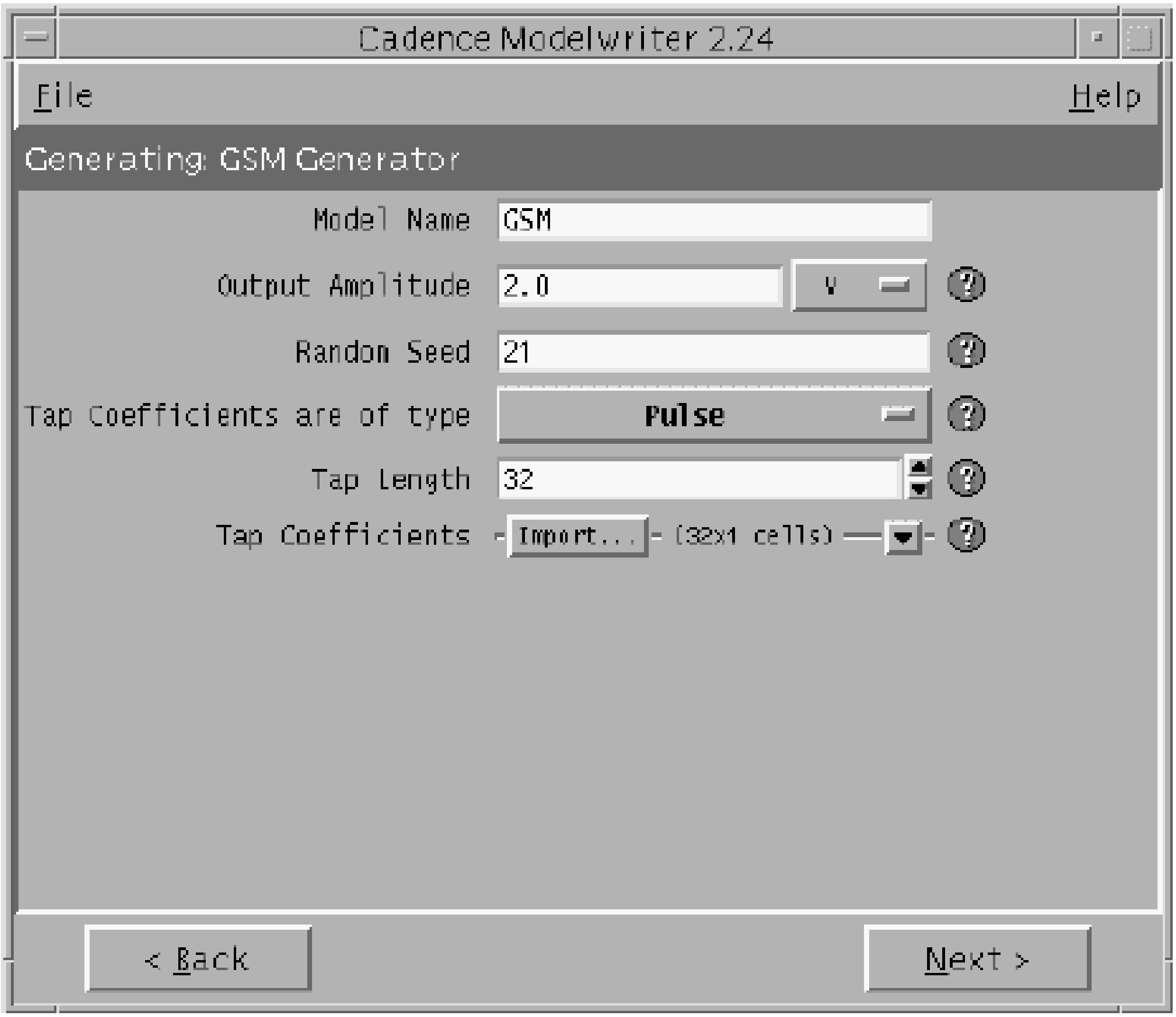
    b. Select the GSM generator.

    c. Click the *Next* button in the lower right hand corner of the Modelwriter window.

    You should now see the picture in Figure 1-125 in the window.

© 1994-2020 All Rights Reserved.

Figure 1-125 GSM Generator



2. Specify how you plan to drive the filter by selecting the type of tap coefficients.

   FIR filters can be driven by pulses or impulses.

3. Specify the length of the FIR filter in the *Tap Length* field.

4. Specify the tap coefficients, in one of the following ways.

   - From a file
   - By direct manual entry.

   To read the coefficients from a file do the following:

© 1994-2020
All Rights Reserved.

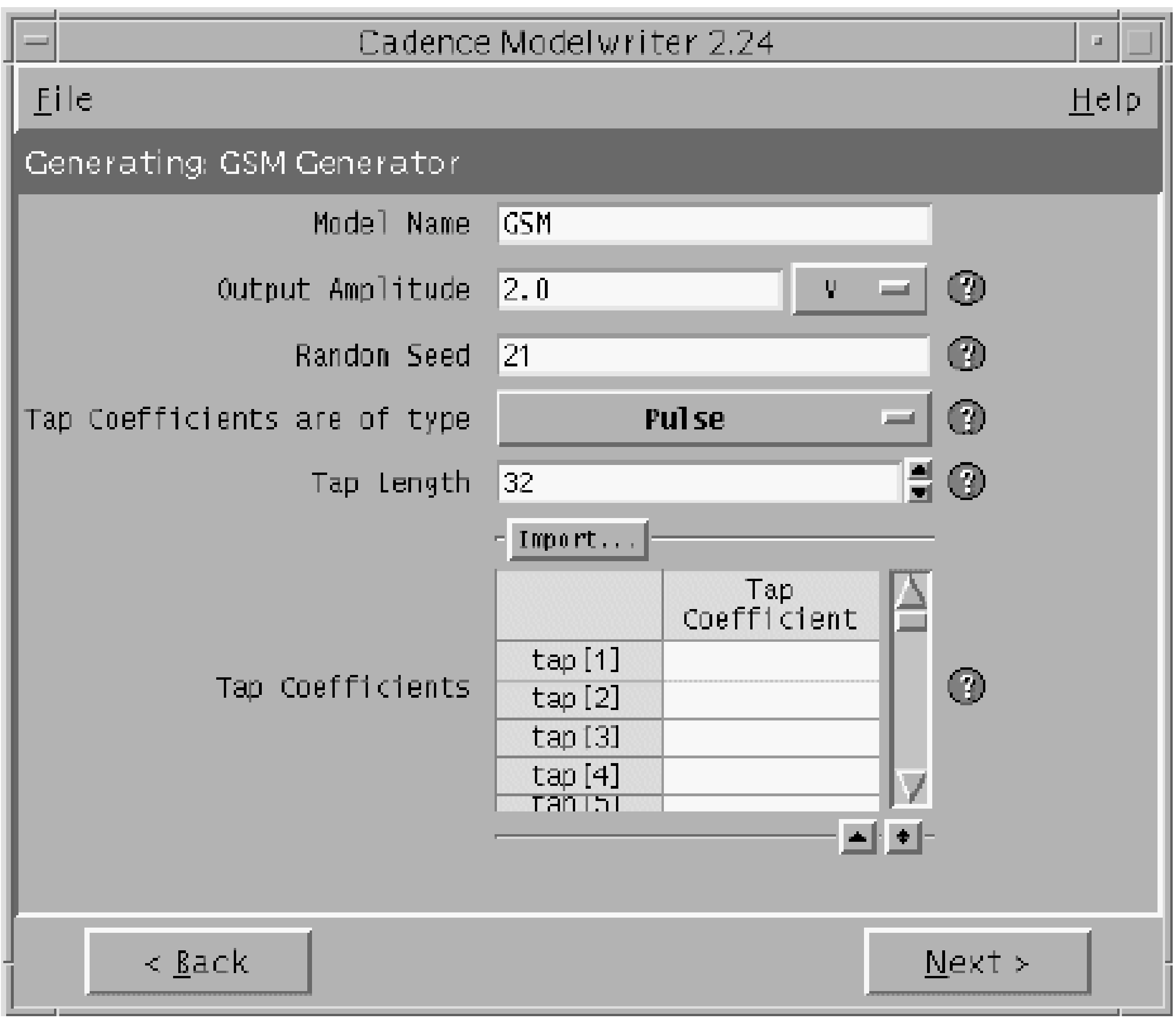
a. Select the *Import* button.

b. Enter the path to the file.

c. Click *Open*.

The coefficients appear in the window as shown in Figure 1-126. (The tap[1] coefficient multiplies the filter state with the least amount of delay, the filter state closest to the input.)

Figure 1-126 Tap Coefficients



To enter the coefficients manually do the following:

© 1994-2020
All Rights Reserved.

a. Click the lower rightmost button flagged in the form.

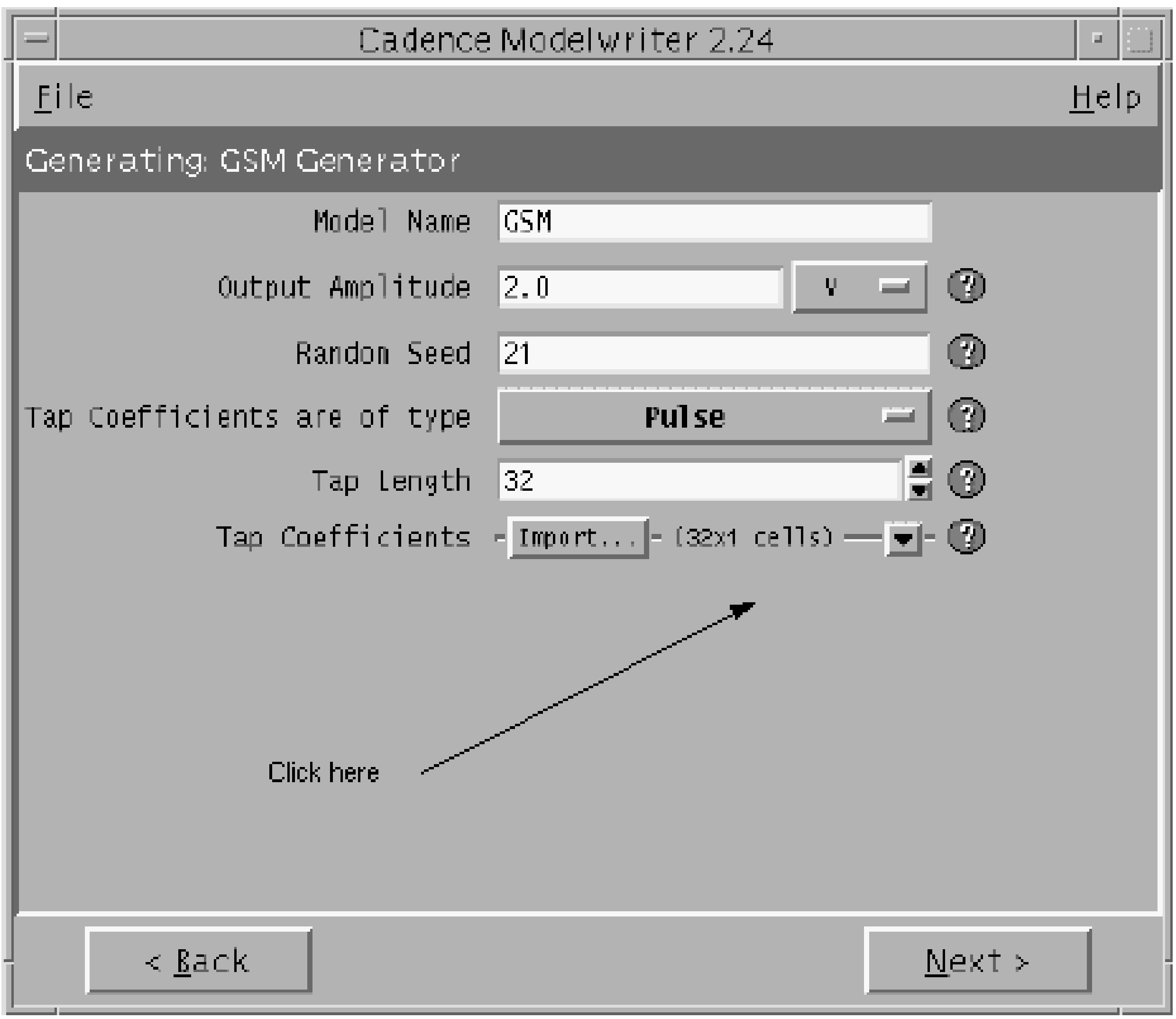
See Figure 1-127.

b. Enter the coefficients,

c. Click *next* to view the model,

5. To write the model to a file, click *Save Generated Code...*.

Figure 1-127 Manual Entry of the Tap Coefficients



© 1994-2020
All Rights Reserved.

© 1994-2020 All Rights Reserved.

2

rfTlineLib Library

This chapter contains the following topics:

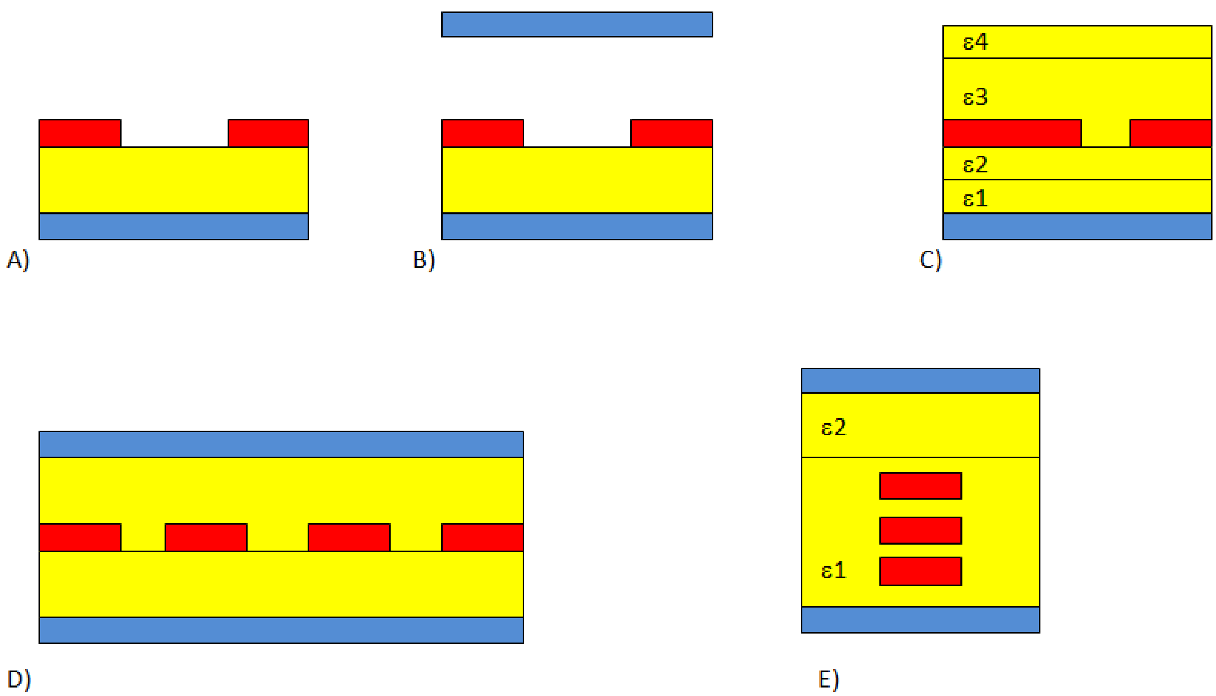
- Introduction to RFTLINELIB on page 300
- Installation on page 302
- Library Structure on page 303
- Quick-Start Guide on page 304
- Using RFTLINELIB on page 310
- RFTLINELIB Cells on page 322
- Microstrip Components on page 380
- Stripline Components on page 391
- Obsolete Components on page 379

© 1994-2020 All Rights Reserved.

Introduction to RFTLINELIB

RF transmission line library `rfTlineLib` is a library of wideband-accurate transmission line models in multi-conductor microstrip and stripline configurations. Based on frequency-dependent per-unit-length parameters calculated by a 2D quasi-static electromagnetic solver, the models are integrated in Virtuoso ADE and are accessible from stand-alone Spectre netlists. `rfTlineLib` includes an interactive graphical stackup editor for storing substrate geometry and material properties.

`rfTlineLib` includes an interactive graphical stack-up editor for storing substrate geometry and material properties. It supports a virtually unlimited range of multi-layer, multi-conductor transmission line configurations, including microstrip, stripline, edge- and broadside-coupled lines, and others. Some examples of transmission line structures supported by `rfTlineLib` are shown below:



Examples of supported transmission line configurations. A) coupled microstrip B) coupled microstrip with a shield C) asymmetric coupled microstrip in inhomogeneous dielectric D) edge-coupled 4-conductor stripline E) broad-side coupled line in inhomogeneous dielectric

© 1994-2020 All Rights Reserved.

RFTLINELIB Benefits

Accuracy

`rfTlineLib` transmission line models, such as `mlin` and `slin`, are based on rigorous EM simulations and include state-of-the-art descriptions of dielectric and conductor losses. Their applicability extends to long transmission lines and wide frequency ranges. The models have been tested extensively against standard industry benchmarks and 3D EM models, matching them within 1% or better.

The runtime simulation performance of new `rfTlineLib` models is comparable to empirical transmission line models and low-order lumped approximations. The use of EM-based transmission line parameters means that, in general, the initial extraction takes considerably longer, a few seconds in typical cases. However, instance and cross-section based caching makes EM extraction a one-time cost. As a result, in typical use, the overall simulation performance remains highly efficient.

Ease of Use

`stackup`, a new cell in `rfTlineLib`, defines common substrate properties. `stackup` offers several pre-configured and commonly used layer stacks and the ability to define custom substrates behind an easy to use, graphical editor. Transmission line cells reference the common substrate definition without the need to re-enter shared parameters.

Smart Discontinuities

rfTlineLib discontinuities - tees, bends, curves and others - do not require users to specify redundant connectivity information. Instead, the properties of adjacent lines are collected automatically from the circuit's topology.

RFTLINELIB Cells Vs. analogLib Mtline

`rfTlineLib` transmission lines are based on Spectre `mtline` technology and are netlisted as instances of `mtline`. In that sense, `rfTlineLib` models may be viewed as features of `mtline` and not a substitute. `mtline` has been enhanced significantly to support `rfTlineLib`. These enhancements include:

- Improvements to EM solver accuracy. The solver is now within 1% of the established benchmarks.

© 1994-2020 All Rights Reserved.

- New Spectre primitives -- including stackup, dielectric, and conductor - and new `mtline` parameters, to support a common substrate definition and the ability to reference it from mtline.
- New frequency-dependent models of dielectric losses, including Debye, Wideband Debye, and table-based models.
- The Hammerstad surface roughness model for enhanced modeling of conductor losses.
- EM solver speed improvements.

In most cases, users will find `rfTlineLib` models more intuitive and convenient than `analogLib` `mtline`. Nevertheless, there are several use scenarios in which `analogLib` `mtline` is the only choice: they include coplanar waveguide configurations and the use of RLGC-form input as transmission line parameters.

Limitations

- Discontinuity models are not supported in shooting envelope and shooting-based pss.
- Discontinuities are modeled as short-circuits in transient.
- rfTlineLib does not support coplanar-waveguide configurations.

Installation

Location

`rfTlineLib` is located in the `$(compute:THIS_TOOL_INST_ROOT)/tools/dfII/samples/artist/rfTlineLib` directory.

Specify the following in your `cds.lib` file or include it in the library path editor:

```
DEFINE rfTlineLib $(compute:THIS_TOOL_INST_ROOT)/tools/dfII/samples/artist/
rfTlineLib
```

Version Requirements

`rfTlineLib` requires MMSIM 12.1.1 and IC 6.1.6 ISR1 or later.

© 1994-2020 All Rights Reserved.

Library Structure

`rfTlineLib` is organized in categories for easier access and browsing. Legacy models including `microbend2`, `microbend90`, `microopenend`, `microstep`, `microstrip`, `stripbend90`, and stripline have been moved to category *Obsolete* and remain supported for backward compatibility but are otherwise not recommended.

New transmission lines are in the category *TransmissionLines* and presently include the following cells:

- `maclin`: asymmetric 2-conductor coupled microstrip
- `mclin`: coupled 2-conductor microstrip
- `mlin`: single-conductor microstrip
- `sclin`: coupled 2-conductor stripline
- `slin`: single-conductor stripline
- `nclin`: n-conductor transmission line

Discontinuities:

- `mtee`: microstrip T-junction
- `mbend`: mitered arbitrary angle microstrip bend
- `mbend2`: mitered 90 degree microstrip bend
- `mcorn`: microstrip corner
- `mcros`: microstrip cross
- `mstep`: microstrip step
- `stee`: stripline T-junction
- `sbend`: mitered, arbitrary angle stripline bend
- `sbend2`: mitered 90 degree stripline bend
- `scros`: stripline cross
- `sstep`: stripline step
- `mcurve`: microstrip curve
- `scurve`: stripline curve

© 1994-2020 All Rights Reserved.

- `mloceff`: micsrostrip open-circuit effect
- `sloceff`: stripline open-circuit effect

Important

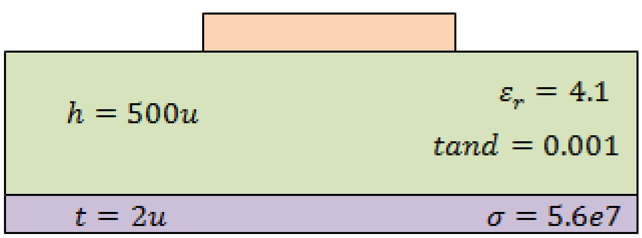
`stackup` is a new cell used to define common substrate parameters. `stackup` is not categorized.

In addition to stackup-based transmission lines and discontinuities, rfTlineLib includes behavioral transmission lines in category Behavioral:

- `tlinp`: lossy transmission line
- `clin`: ideal coupled transmission line

Quick-Start Guide

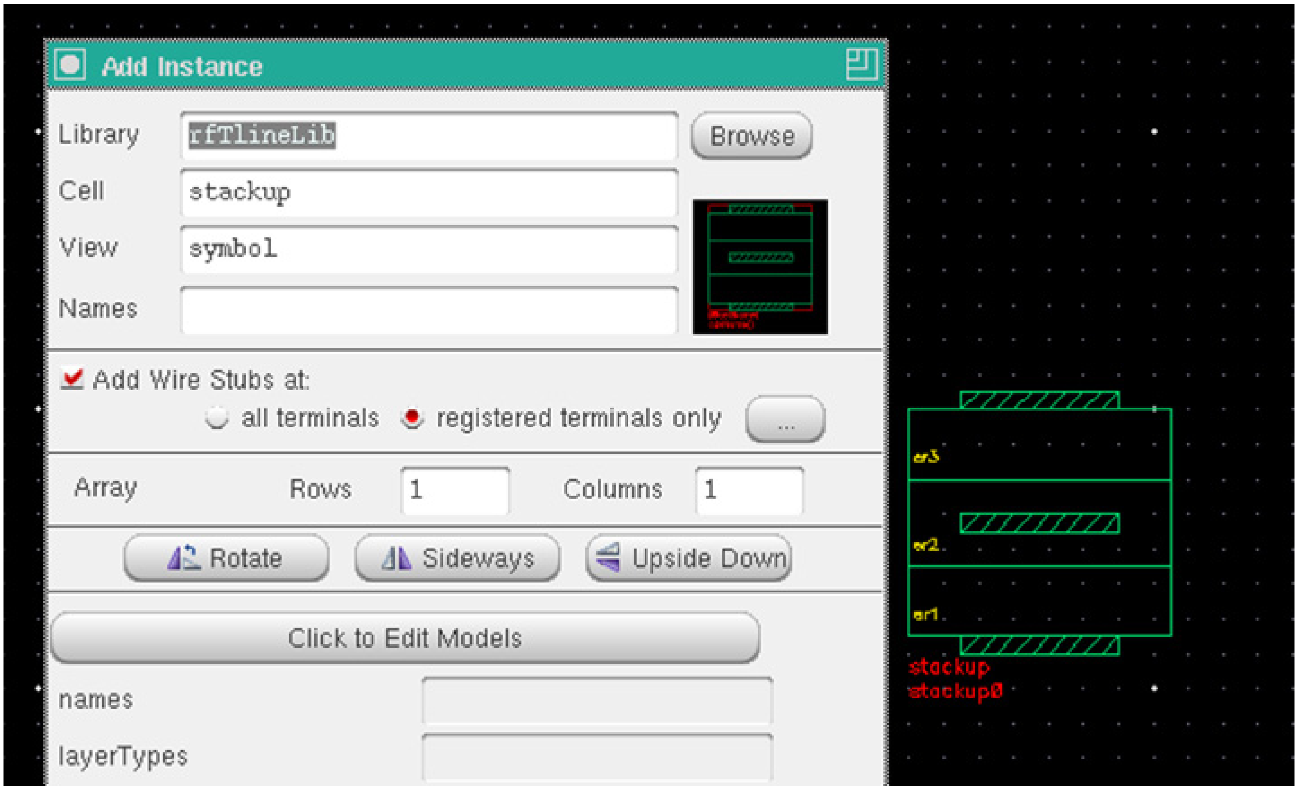
The new `rfTlineLib` features are best illustrated by use of an example. Consider a microstrip trace on 500u FR4, with 2u copper traces and ground plane, as shown below.



Step 1: Insert a stackup object

© 1994-2020 All Rights Reserved.

To define this stack, begin by placing a stackup instance on the schematic, as shown below.



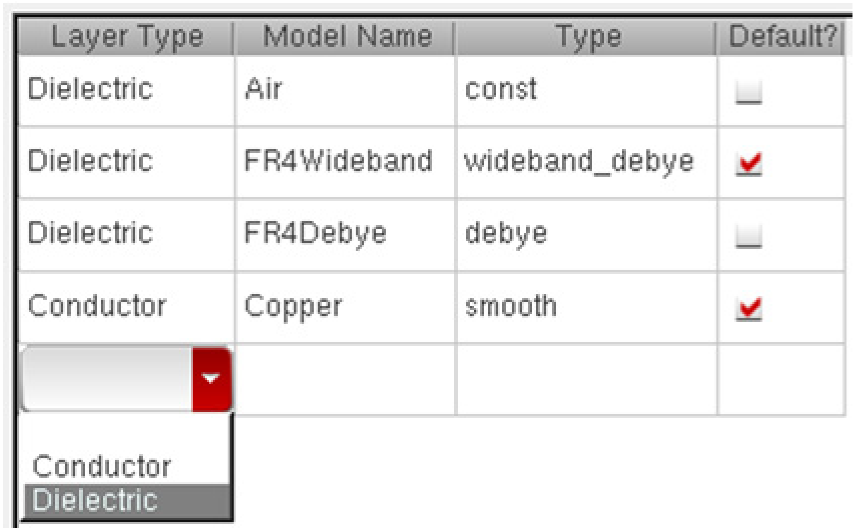
Transmission lines will reference the stackup (shown later). Any number of stackup objects may be present, although in practice, one is usually sufficient. Stackups may be placed anywhere in the hierarchy.

Step 2: Define material properties

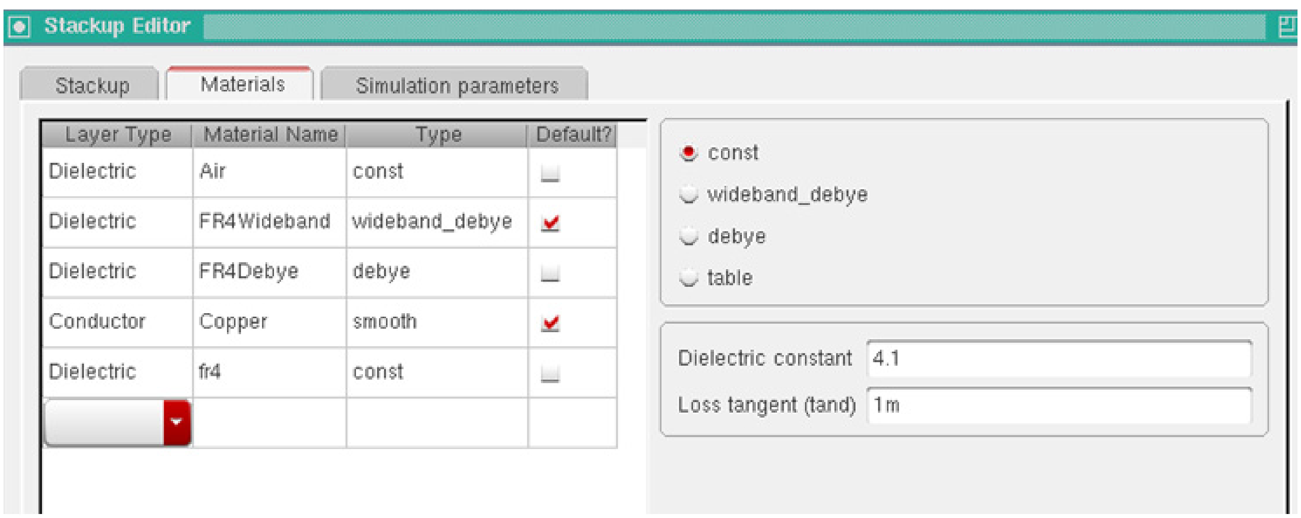
Click the button labeled *Click to Edit Models* on the *Add Instance* form to edit the stack, and click the *Materials* tab. This is where you set the material properties of conductors and dielectrics.

© 1994-2020
All Rights Reserved.

Select a new dielectric from the drop-down list, as shown below.



Name the material fr4, choose the constant loss model, and let the relative permittivity and loss tangent equal 4.1 and 0.001, respectively. The material database should look as follows:



Step 2.1: Setting default materials

© 1994-2020 All Rights Reserved.

If you want fr4 to be the default material when adding new dielectric layers to the stack, select the check box in the *Default* column for fr4, as shown below:

Layer Type	Material Name	Type	Default?
Dielectric	Air	const	☐
Dielectric	FR4Wideband	wideband_debye	☐
Dielectric	FR4Debye	debye	☐
Conductor	Copper	smooth	☑
Dielectric	fr4	const	☑

Now, newly added conductors will use Copper, and dielectrics will use `fr4` by default.

Step 3: Define the layer stack

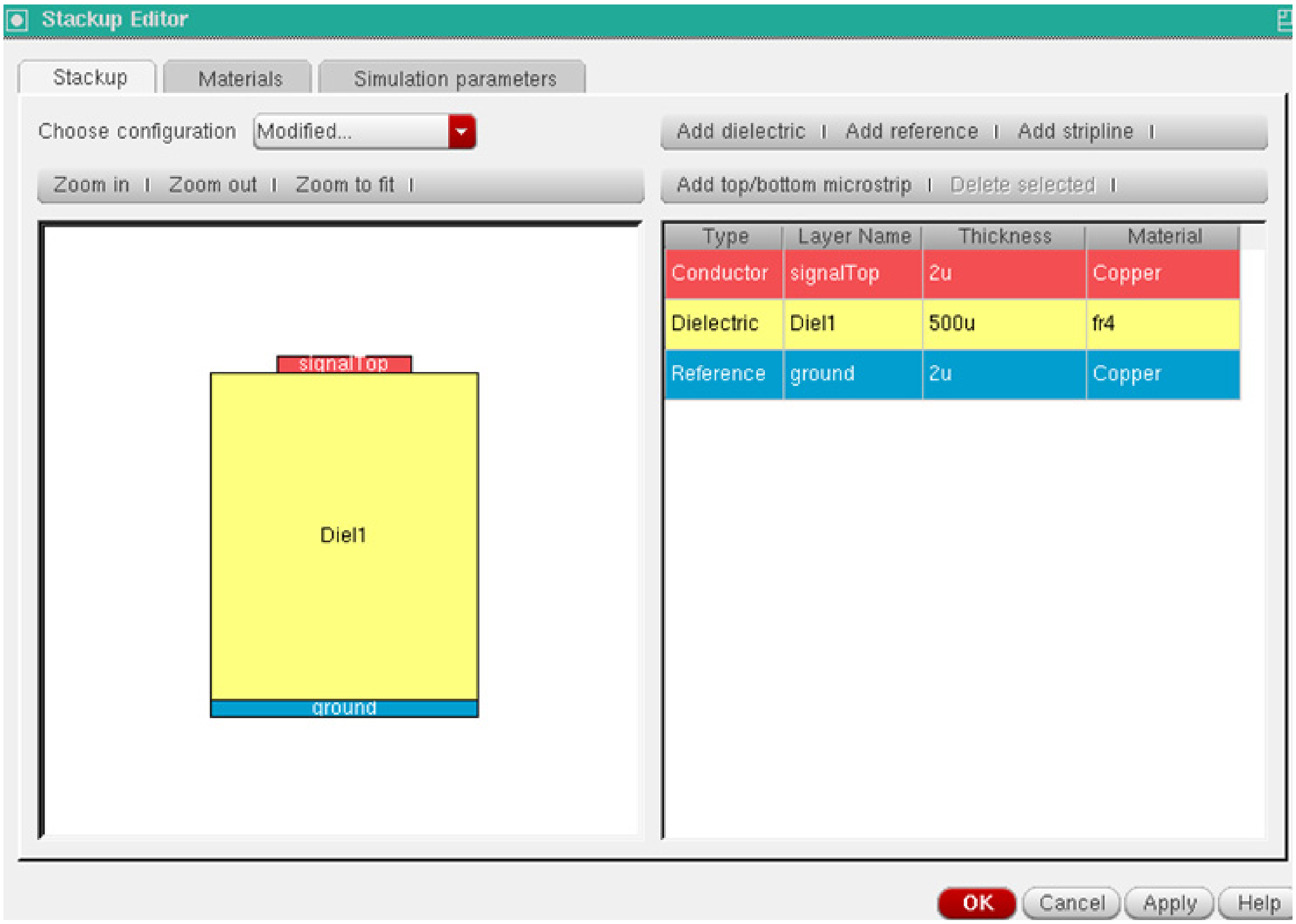
Next, click the Stackup tab to define the layer stack. In our case, we can choose the Microstrip configuration, the Microstrip/Stripline configuration, or we can enter our own configuration from scratch. In this step, we will define our own configuration to illustrate the process of defining a layer stack.

Select *Modified* from the *Choose Configuration* drop-down list, and enter three layers as follows:

- Click *Add reference*, name it *ground*, set the thickness to 2 microns (2e-6 or 2u), and make sure that it uses *Copper* as the model.
- Click *Add dielectric*, leave the name at default, and set the thickness to 500u. Select fr4 by double-clicking on the *Material* cell and choosing *fr4*.
- Click *Add top/bottom microstrip*, set the name to *signalTop*, let the thickness be 2u, and make sure that the *Material* cell specifies *Copper*.

© 1994-2020 All Rights Reserved.

The layer stack should now look as follows:



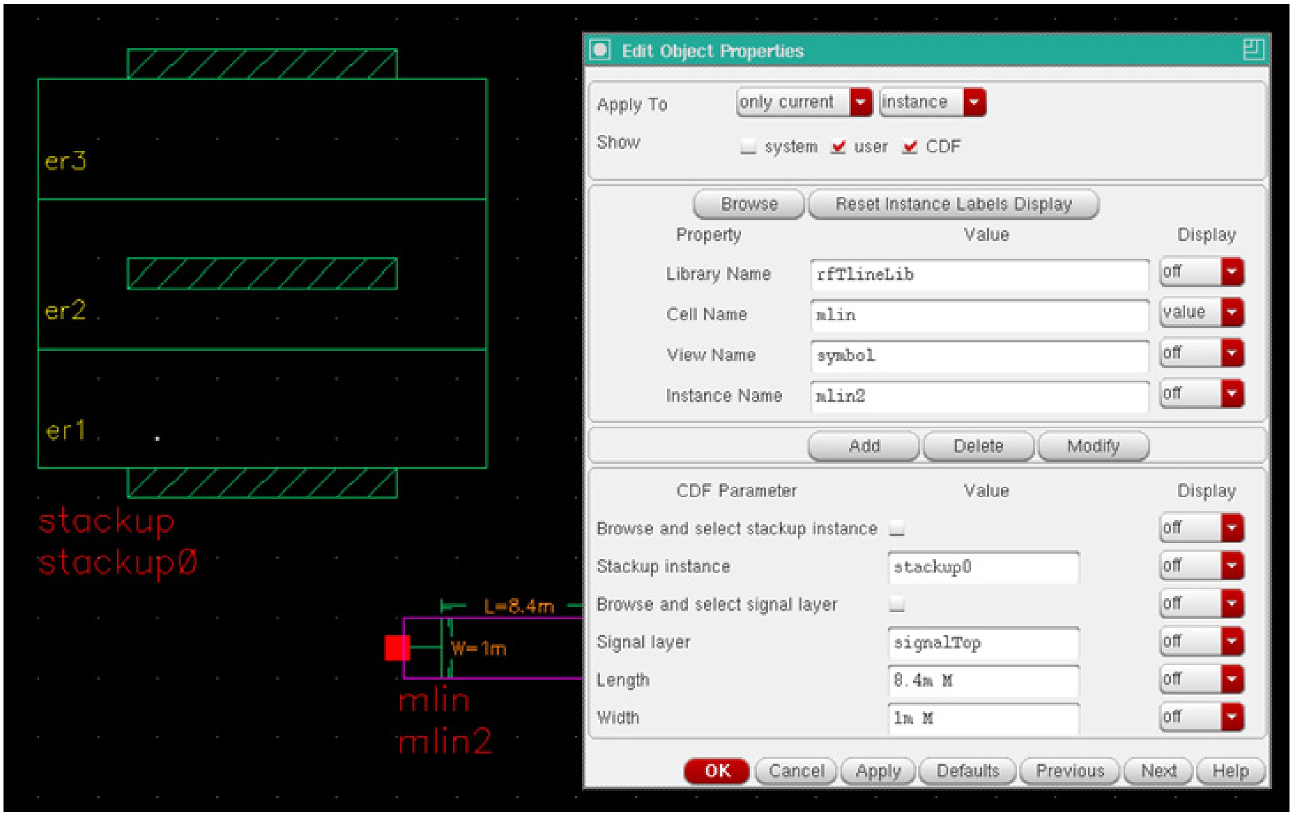
Click *OK* and then *OK* again to exit the CDF form. The stackup is now ready for use.

Step 4: Insert a transmission line

Insert a microstrip line as shown below, making sure that it references the appropriate stackup instance (`stackup0` in our case) and the layer name of the microstrip signal conductor (`signalTop` in our case). Ground layers are identified automatically and are not parameters of transmission line cells. Note the existence of layer and signal name browsers on the CDF form. If the stackup object is placed on the same schematic as the lines, you can use the browsers for easy specification of layer and stackup names. Clicking *Browse* and selecting *stackup instance* displays the names of the stackup instances on the schematic.

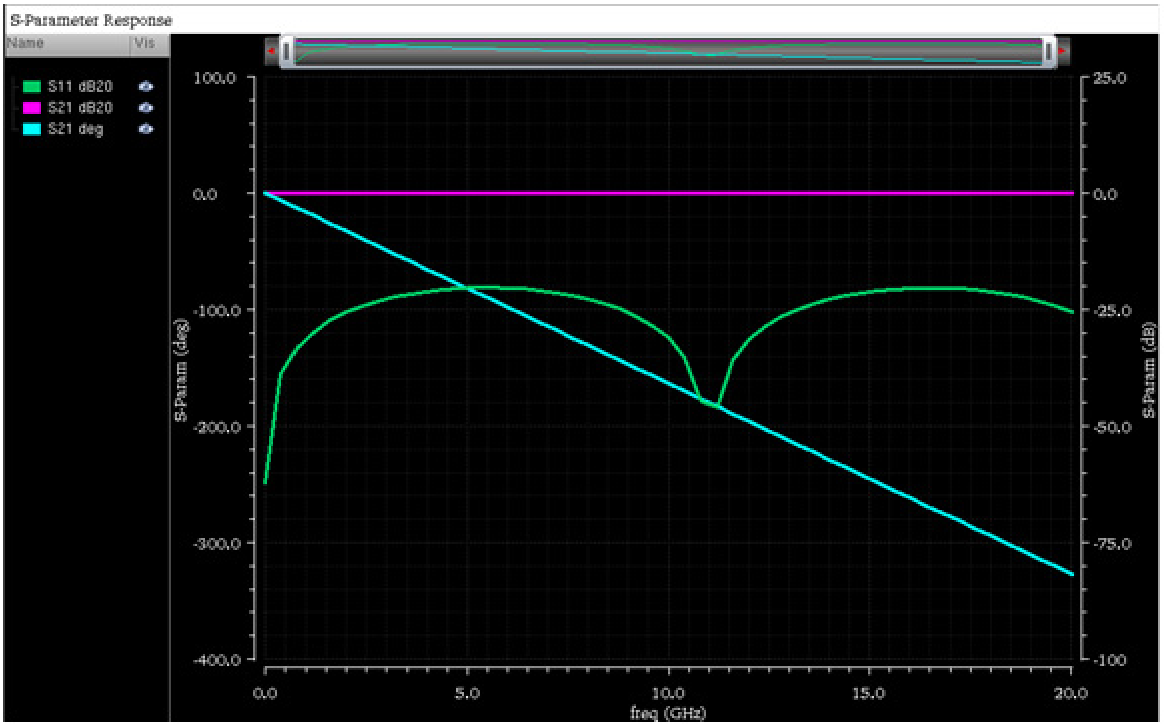
© 1994-2020 All Rights Reserved.

Similarly, for a given stackup instance, browsing and selecting signal layer lists all its signal conductors.



Let the width and length be equal to 1 millimeter (1m) and 8.4m, respectively. The microstrip line is ready to be used in the simulation. Simulate 50-Ohm S-parameters from DC from 20 GHz to obtain the following response:

© 1994-2020
All Rights Reserved.



Using RFTLINELIB

The Stackup

`rfTlineLib` transmission line cells reference stackup objects. Stackup objects store material properties and cross section information. Stackups may be placed anywhere in the hierarchy and are specified by their Spectre-style hierarchical paths.

Dielectric Models

`rfTlineLib` offers four methods of modeling frequency dependence of relative permittivity $\varepsilon_r(\omega)$:

- Constant ε_r and loss tangent (tan δ)
- Wideband Debye model as proposed by *Djordjevic-Sarkar [Ref. 1]* and *Svensson-Dermer [Ref. 2]*
- Multipole Debye model

© 1994-2020
All Rights Reserved.

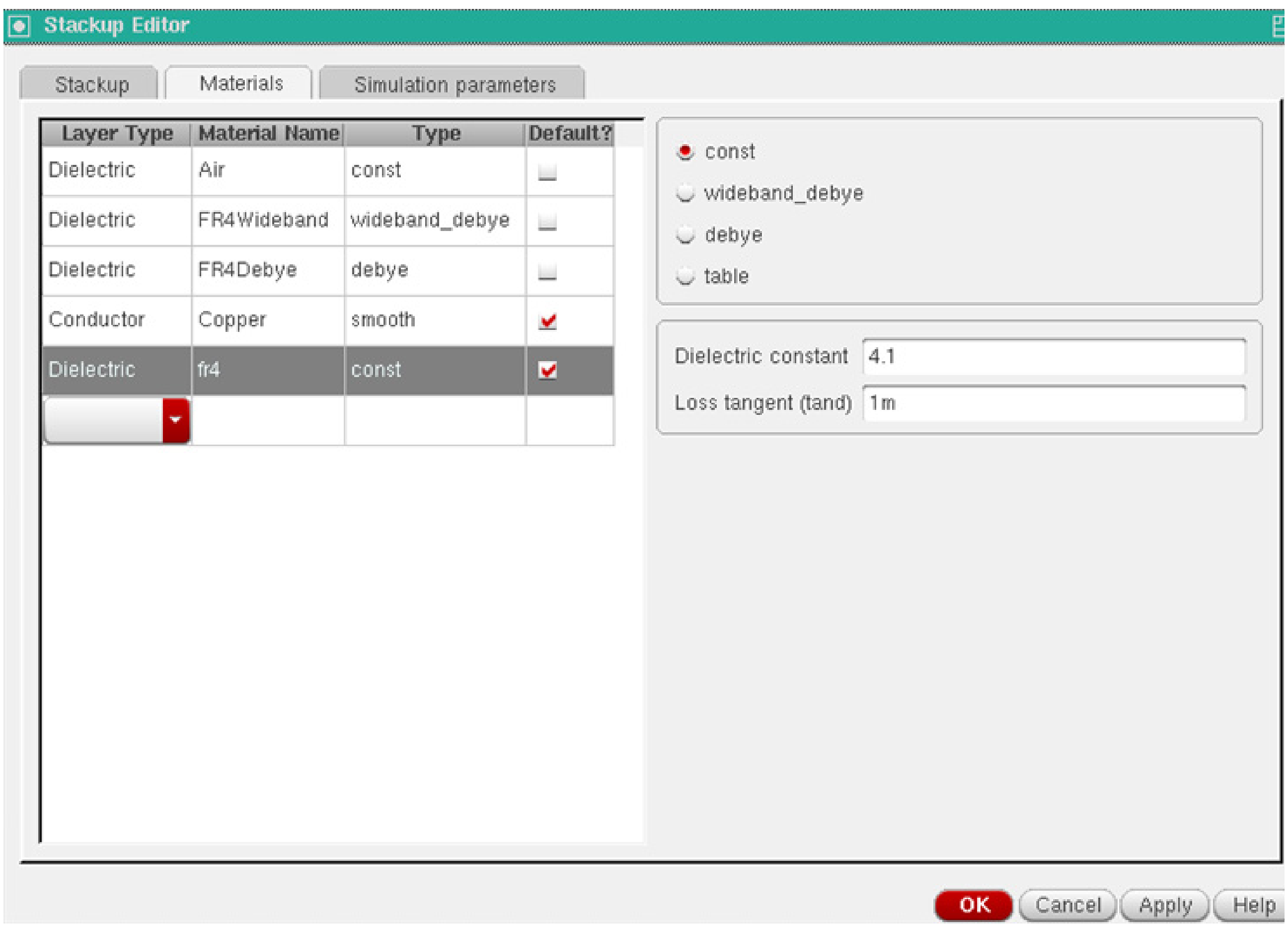
- Table model that constructs the multipole Debye model from the specified table of frequency-dependent dielectric constant and tangent delta

Constant Model

The constant model assumes no frequency dependence. In particular, the relative permittivity $\varepsilon_r(\omega)$ is of the form:

$$\varepsilon_r(\omega) = \varepsilon_r'(1 - j \bullet \tan\delta)$$

You specify the constant model by setting the dielectric constant ε'_r and the loss tangent $\tan\delta$ in the stackup object, as shown below.



The constant model is the oldest and most commonly used in the frequency domain. A drawback of this model is that it is non-causal if $\tan\delta \neq 0$. In practical use, the constant

© 1994-2020
All Rights Reserved.

dielectric model is recommended for dielectrics with very low loss, such as Duroid, Silicon dioxide ($SiO2$), and so on. The largest loss tangent that should be used in this entry is 0.005. Since many commercial EM simulators implement the constant dielectric model, it may also be useful for comparison and verification.

Wideband Debye

The Wideband Debye model expands the relative permittivity as [Ref. 1, 2].

$$\varepsilon_r(\omega) = \varepsilon_{r,\infty} + \frac{\delta\varepsilon_r}{(m_2 - m_1)\ln(10)}\ln\left(\frac{\omega_2 + j\omega}{\omega_1 + j\omega}\right)$$

The model is constructed from the following user-specified data:

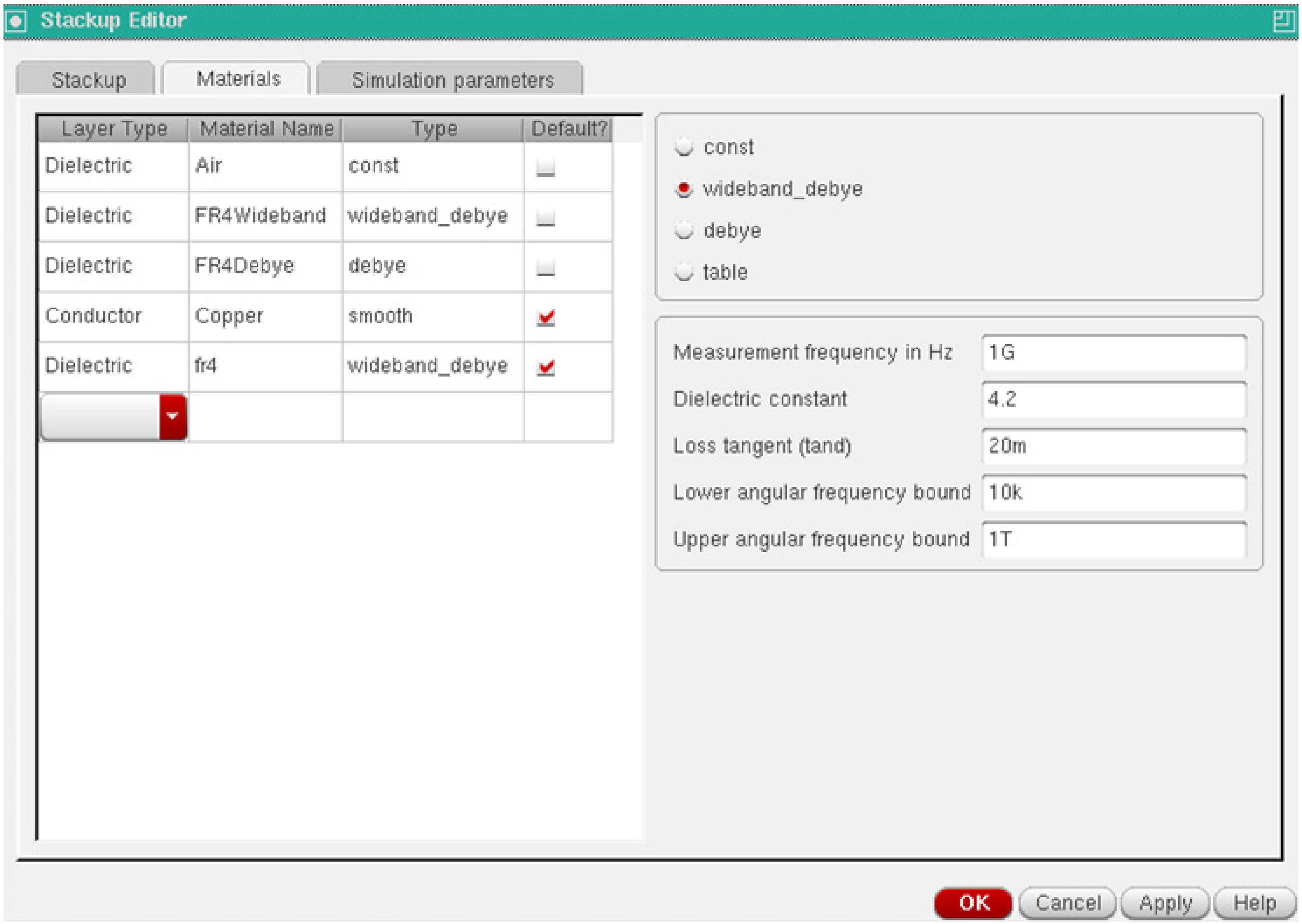
- The measurement frequency
- The dielectric constant and the loss tangent at the measurement frequency
- ω_1 and ω_2, angular frequency bounds, whose typical values are 1-1e4 rad/s and 1e10 to 1e12 rad/s, respectively;

Wideband Debye is a causal model for frequency-dependent dielectric constant of lossy dielectrics. It is constructed from the values of the dielectric constant and the loss tangent at one frequency point, so it combines the simplicity of the constant model with the benefit of causal response. The other two parameters (ω_1 and ω_2) are the lower and upper bound of the angular frequency where the loss tangent $\tan\delta$ varies approximately linearly with frequency. The model is not very sensitive to these values, so setting ω_1=100 and ω_2=1e12 is reasonable. This model is most commonly used for the FR-4 dielectric and pre-preg utilized to manufacture common PCBs. It is recommended for the modeling of FR-4 PCBs.

The model is constructed such that the frequency-dependent dielectric constant and tangent delta of the model matches those given at the measurement frequency, and the loss tangent varies approximately linearly between ω_1 and ω_2. Note that in Spectre the angular frequencies are specified, $\omega_1 = 2\pi f_1$ and $\omega_2 = 2\pi f_2$.

© 1994-2020 All Rights Reserved.

You specify the Wideband Debye model parameters in the *Stackup Editor*, as follows:



Debye

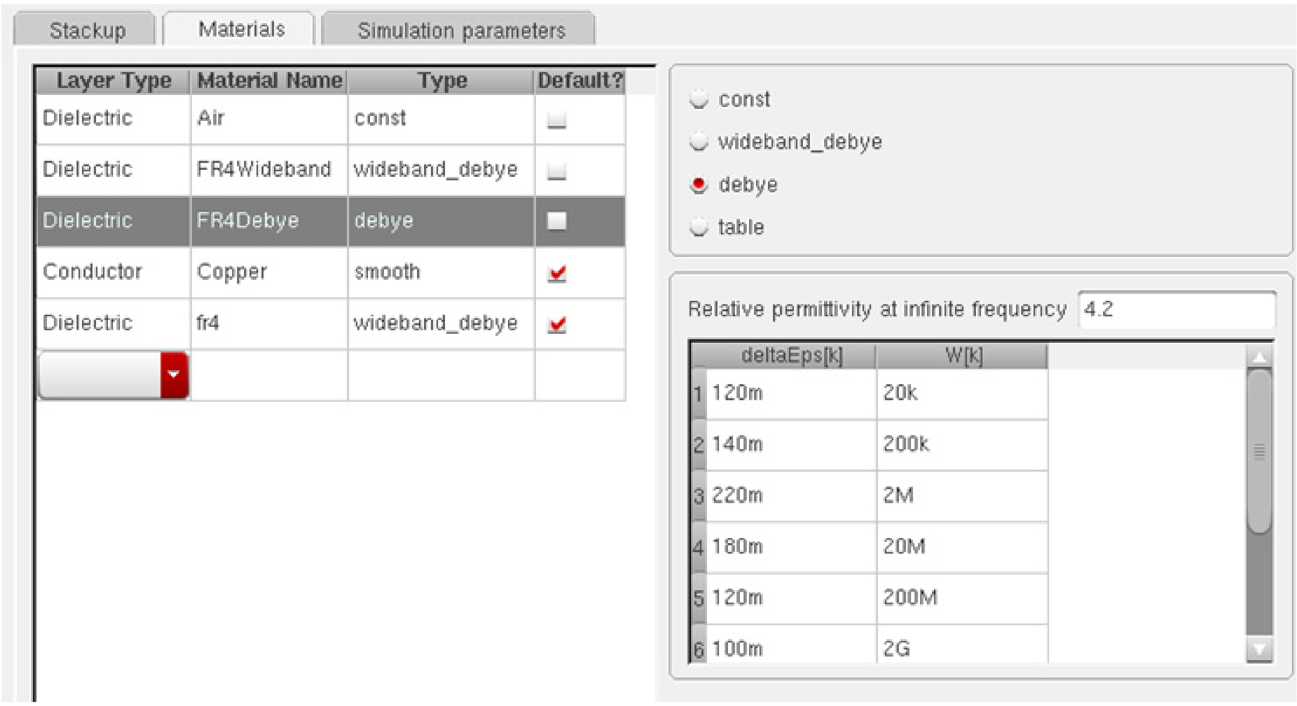
The Debye model uses the following multi-pole expansion:

$$\varepsilon_r(\omega) = \varepsilon_{r,\infty} + \Sigma_{k=1}^{N} \frac{\delta\varepsilon_{r,k}}{1 + j\frac{\omega}{\omega_k}}$$

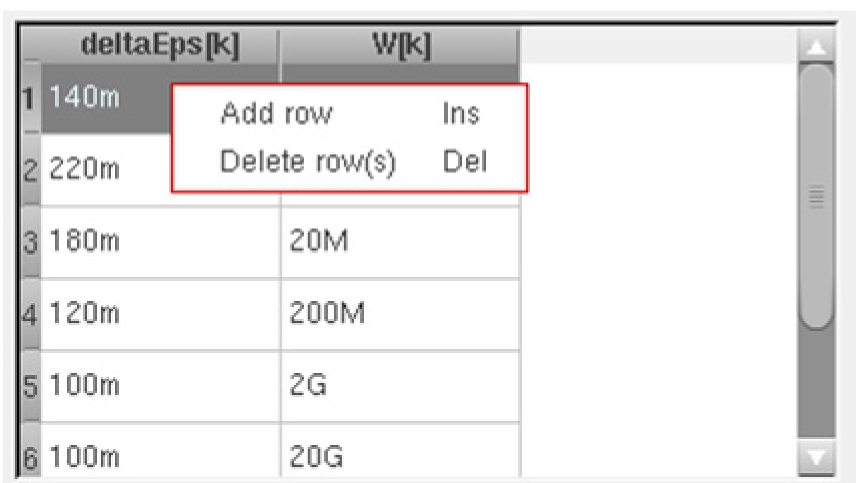
N, $\delta\varepsilon_{r,k}$, ω_k and $\varepsilon_{r,\infty}$ are model parameters and are specified by the user and entered in

© 1994-2020
All Rights Reserved.

the *Stackup Editor*, as shown below.



To add or delete a row of data, select the row, right-click, and select *Add row* or *Delete row(s)* from the context menu. Alternatively, use the bindkeys, as shown below.



Multipole Debye is recommended for the dielectrics for which the Wideband Debye model does not describe the frequency dependence well. It can approximate the experimental data of the frequency dependent dielectric constant more closely than the wideband Debye model provided the experimental results are available at a sufficient number of frequency points.

© 1994-2020
All Rights Reserved.

Note that it is necessary to have at least three frequency points to construct a usable multipole Debye model, and it is recommended to have at least 10 frequency points [Ref. 3].

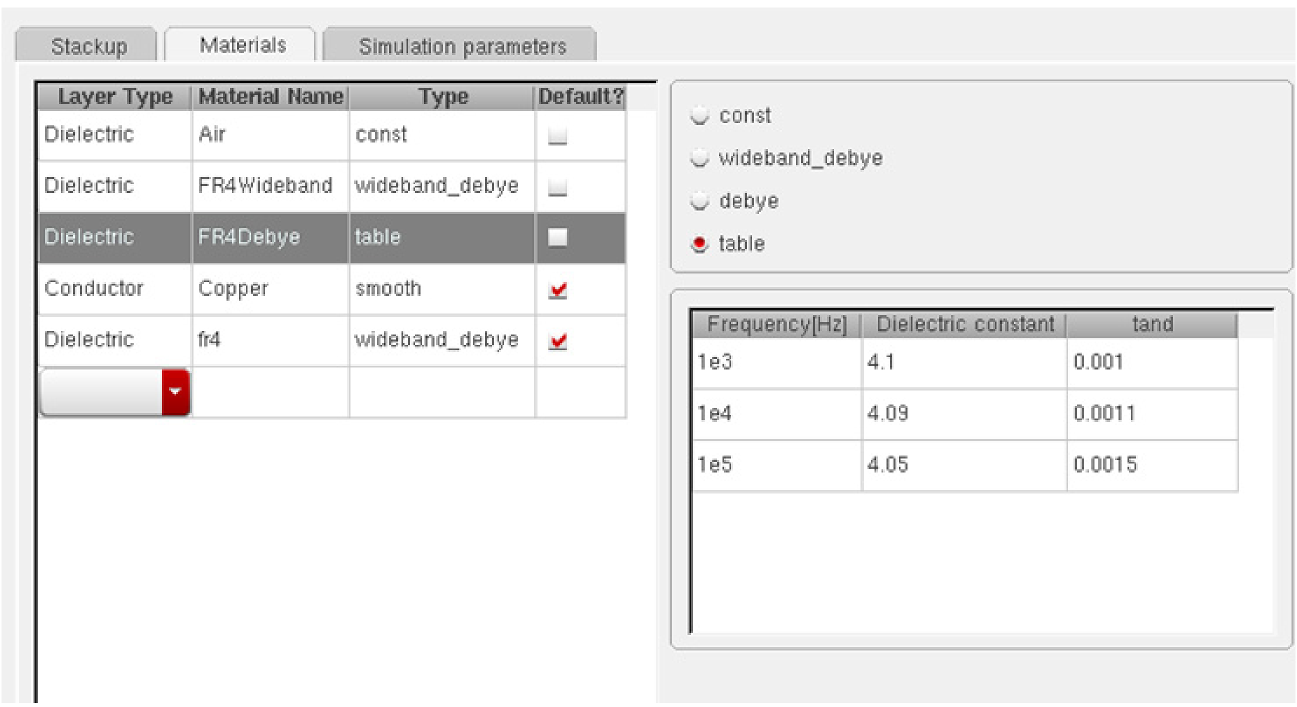
We preload the published multipole Debye model for FR-4 from *Djordjevic-Sarkar's* paper [1].

Table Model

The table model assumes a general frequency-dependence of the relative permittivity.

$$\varepsilon_r(\omega) = \varepsilon_r'(\omega)(1 - j \bullet \tan\delta(\omega))$$

You can specify $\varepsilon_r'(\omega)$ and $\tan\delta(\omega)$ in tabular format, as shown below.

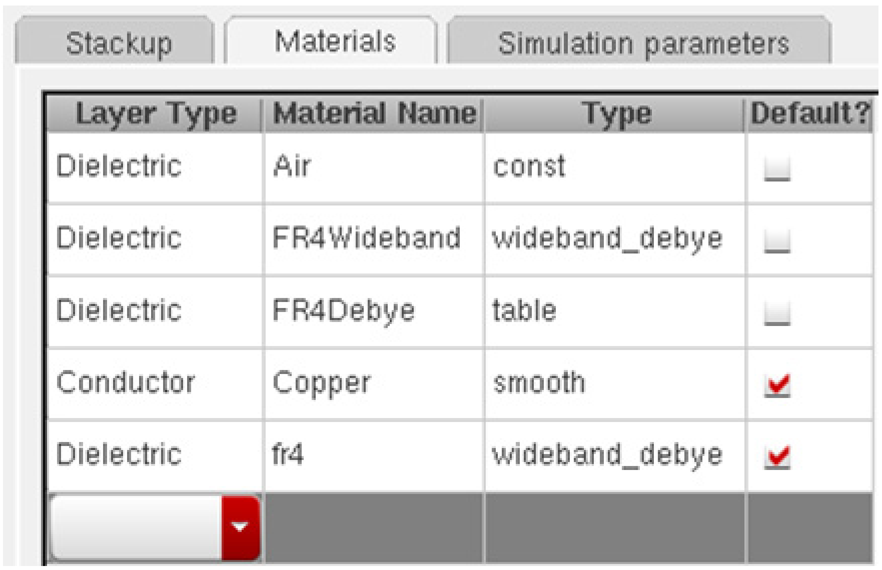


You can add and delete rows by selecting a row and right-clicking or by using Insert and Delete bindkeys, similar to the Debye model.

© 1994-2020
All Rights Reserved.

Adding Dielectric Material Models

The material browser is pre-configured with several default materials. You can add as many materials as you wish by using the drop-down menu at the bottom of the model table:



To change the name of a material, double-click the *Material Name* cell. Default materials are those that are assigned automatically to the newly added layers. A material is designated as a default material by selecting the *Default?* check box.

Conductors

Conductors are specified in terms of their conductivity σ in S/m. One can model smooth conductor surfaces (*smooth*), or model the surface roughness included by means of Hammerstad formula [4] (*hammerstad*).

The Hammerstad model introduces the RMS tooth height parameter hRMS (the units of length, typical value in micrometers), and accounts for the increase of the frequency dependent resistance and inductance per unit length through the use of the Hammerstad coefficient.

$$K_H(\omega) = 1 + \frac{2}{\pi} arc\tan\left(1.4\left(\frac{h_{RMS}}{\delta}\right)^2\right)$$

where $\delta = \sqrt{\frac{2}{\omega\mu\sigma}}$ is the skin-depth. The 2-D EM calculations are performed for the

© 1994-2020
All Rights Reserved.

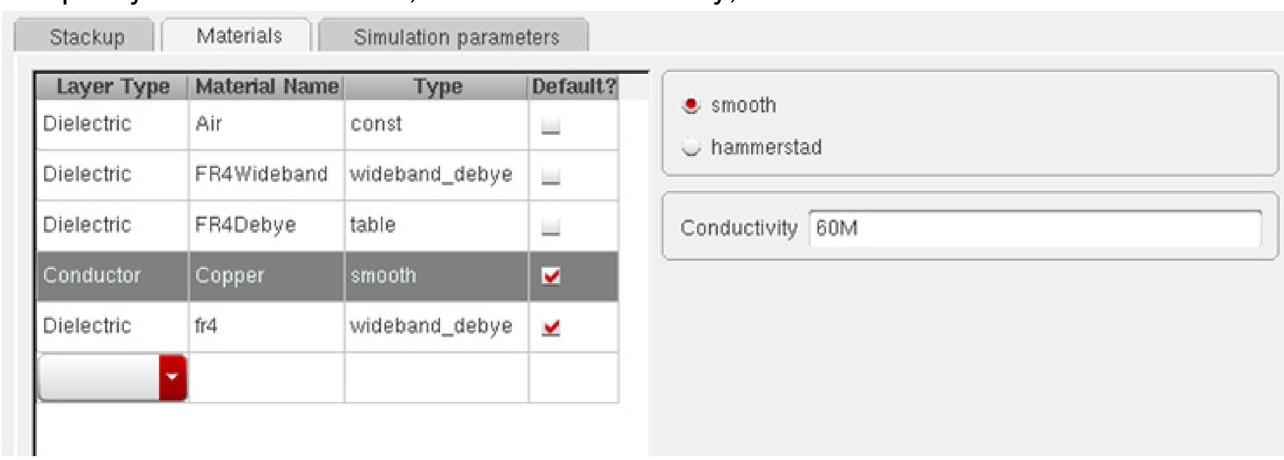
conductors with the smooth surface, and the parameters of the transmission line with rough conductor surfaces, R_H (ω) and L_H (ω) are calculated from those for the smooth conductor surfaces, $R(\omega)$ and $L(\omega)$ according to the expressions below.

$$R_H(\omega) = R_{DC} + K_H(\omega)(R(\omega) - R_{DC})$$

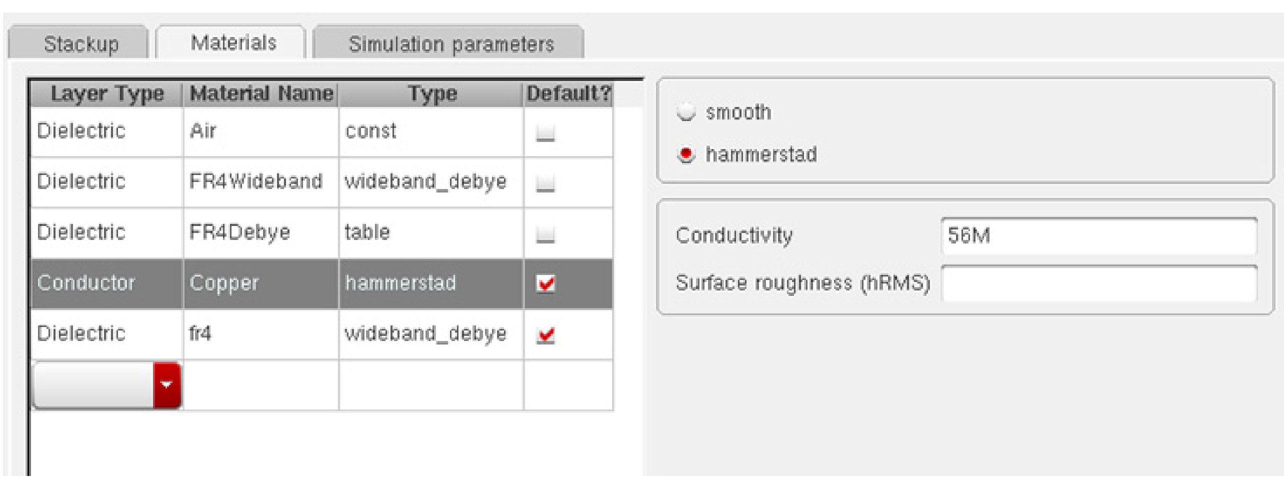
$$L_H(\omega) = L_\infty + K_H(\omega)(L(\omega) - L_\infty)$$

This approach assumes that all conductors have the same roughness.

To specify the smooth model, enter the conductivity, as shown: below.



To include the roughness model, select *hammerstad* and specify the conductivity and RMS height, as shown below.



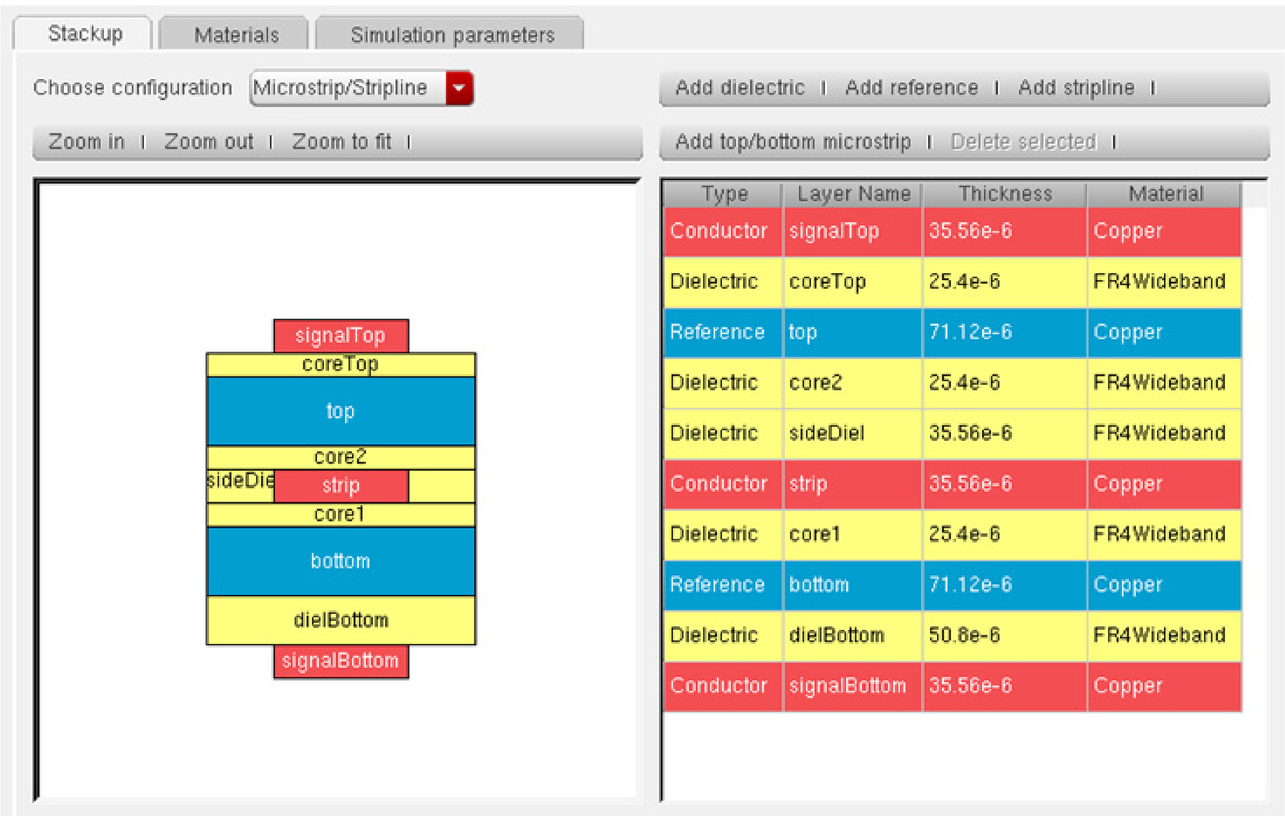
© 1994-2020
All Rights Reserved.

Adding Conductor Materials

You can add as many conductor materials as you wish. The procedure is similar to adding dielectrics.

Configuring Layers

Use the *Stackup* tab to set up your substrate. The *Substrate Editor* is preconfigured with three layer stacks: *Microstrip/Stripline*, *Microstrip,* and *Stripline*. The existing configurations are sufficient to describe many practical scenarios. The *Microstrip/Stripline* configuration is the most general of the three, and is shown below.



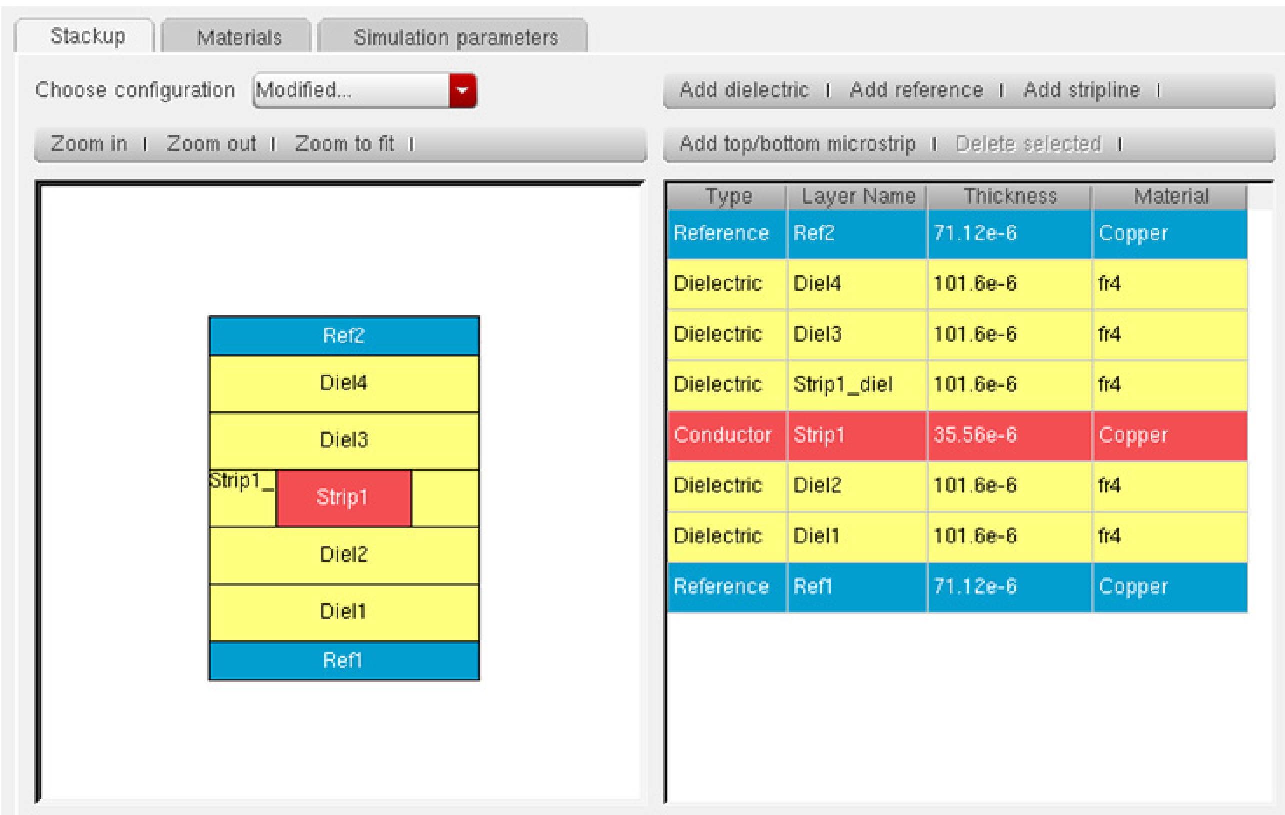
Since *Microstrip/Stripline* already includes two microstrip and one stripline configuration, the standalone Microstrip and Stripline are included mainly as examples. You can freely use the more general stack to define simpler cross-sections. Internally, the solver automatically strips away the unused layers. For example, if a microstrip line were defined on `signalTop`, the solver would ignore everything beneath the top ground plane. Similarly, if a stripline were defined on strip, everything above top and below bottom would be ignored without performance penalty.

© 1994-2020 All Rights Reserved.

To edit layer names, thicknesses or model types, double-click the appropriate cell and edit as appropriate.

Entering Custom Layer Stacks

To enter an arbitrary stack, delete the existing configuration by selecting all layers (click on the top layer in the table or the graphical view, then <SHIFT>+click on the bottom layer, and choose *Delete selected*). Starting from the bottom, enter the appropriate layer one at a time by choosing the appropriate commands using the button strip in the top right corner. The following stack is produced by the sequence of commands: *Add reference*>*Add dielectric*>*Add dielectric*>*Add stripline*>*Add dielectric*>*Add dielectric*>*Add reference*.



Inserting Layers

Layers are inserted by selecting a layer and using one of the *Add layer* commands. New layers are always added on top of the selected stack. If no layer is selected, new layers are added to the top of the stack.

© 1994-2020
All Rights Reserved.

Bindkeys

The following bindkeys are supported:

- Zoom in: <Ctrl>+L
- Zoom out: <Ctrl>+S
- Zoom to fit: <Ctrl>+F
- Delete layers, models or rows in material model tables:<Delete>
- Add rows in material model tables: <Insert>

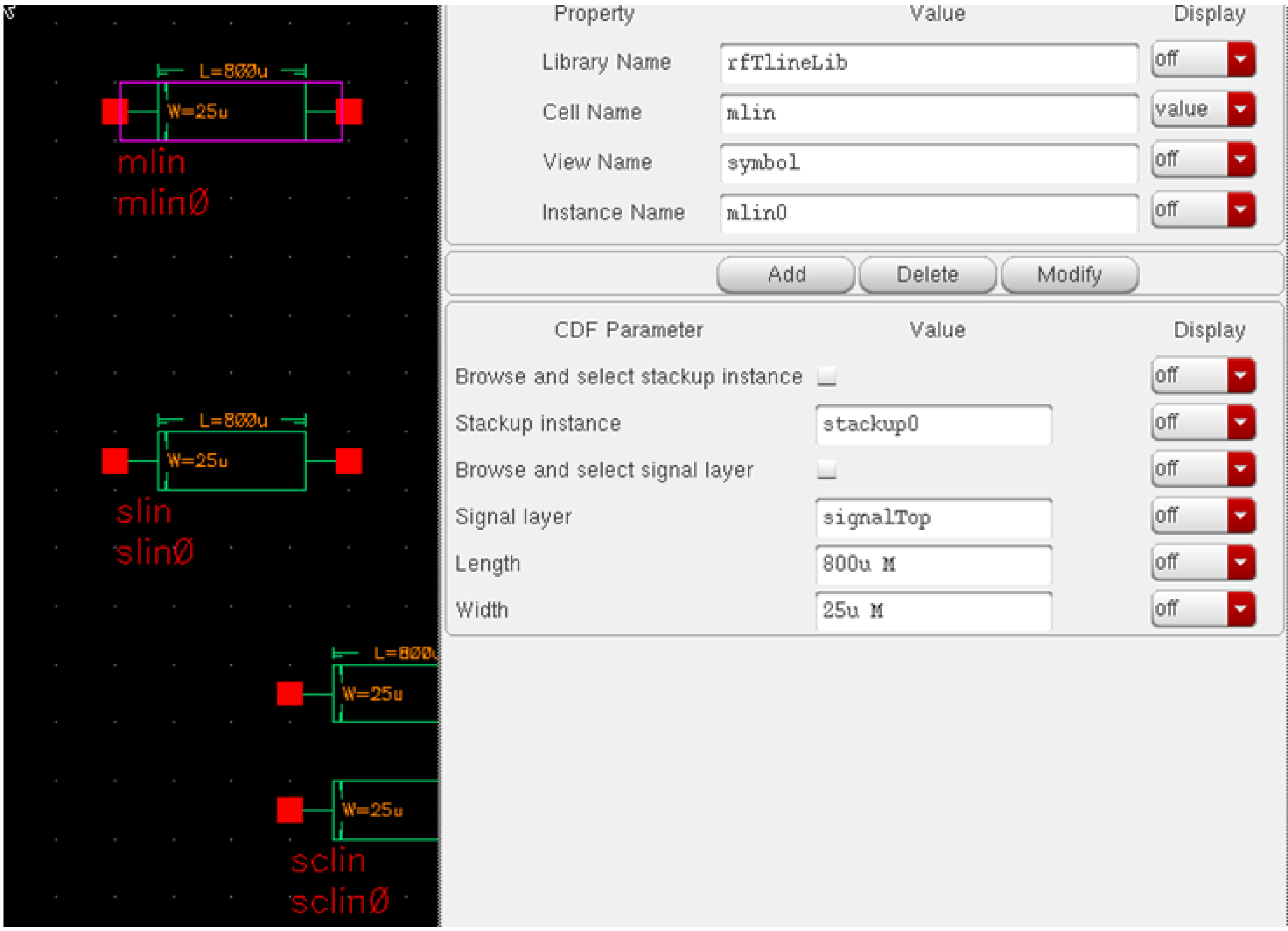
Units

Layer thicknesses are expressed in meters. The use of scale factors, such as p, n, u, m, K, M and G is supported. You can specify thickness in mils but the stack editor will automatically convert them to meters. For example, typing in "2mil" for the thickness of a layer will automatically convert it to 50.8u.

© 1994-2020 All Rights Reserved.

Adding Transmission Lines

Transmission lines reference stackup objects by means of the *Stackup instance* parameter, as shown below.



Stackup instance is a hierarchical instance path in the Spectre language format. Thus:

- *stackup0* matches an instance defined at the current level of hierarchy or any of the parents, in that order of precedence.
- *I0.stackup0* matches a stackup object defined in subcircuit instance I0 defined at the present level of hierarchy or at the top-level circuit, in that order of precedence.

Select the *Browse and select stackup instance* check box to display the stackup browser. Similarly, select the *Browse and select signal layer* check box to display a list of signal layers for the chosen stackup. The browsing aids work only when the stack is defined on the same schematic as the transmission line instance.

© 1994-2020
All Rights Reserved.

Working With Transmission Line Discontinuities

Transmission line discontinuities are in the category *Discontinuities*. Discontinuities have the following important properties:

Automatic Detection of Adjacent Lines

`rfTlineLib` discontinuities are aware of the lines that connect to them and do not require you to specify their geometric and material properties. This information is detected by Spectre at runtime and passed automatically.

Discontinuities Cannot be Simulated Unless They Connect to Transmission Lines

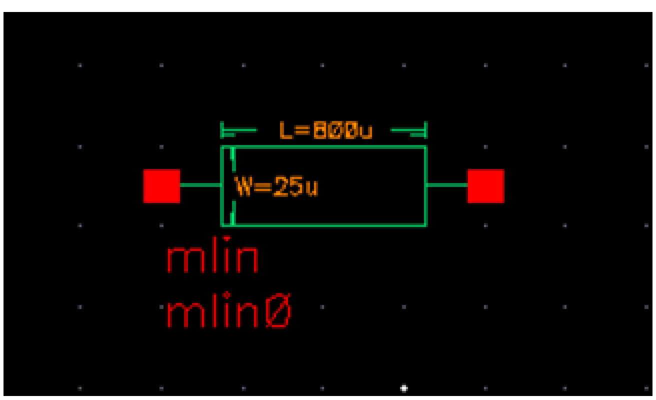
To be simulateable in Spectre, discontinuities must be connected to transmission lines. Leaving them open-circuited or connected to other device types results in Spectre simulation errors.

Discontinuities are Short Circuited in Transient Simulation

In transient and PSS shooting simulations, discontinuities are treated as short-circuits. This is a limitation of the present implementation that will be resolved in future releases.

RFTLINELIB Cells

SYMBOL: MLIN



Microstrip line

Command-line help

```
spectre -h mtline
```

CDF Parameters

Label	Parameter	Type	Units	Default
Stackup Instance	Stackup	Instance name		stackup0
Signal Layer	Layer	String array		signalTop
Length	Len	Double	meter	800u
Width	Linewidth	Double array	meter	25u

Note: `mlin` is not limited to traditional microstrip configurations in the strictest sense of the word - with air above the conductor and a homogeneous dielectric beneath. In fact, the stackup can include any number of dielectric layers of arbitrary material properties, and an optional ground shield above the conductor.

mlin is in effect a general single conductor line above ground (with an optional ground shield). The mlin cell was provided for convenience and for naming compatibility with legacy microwave EDA tools.

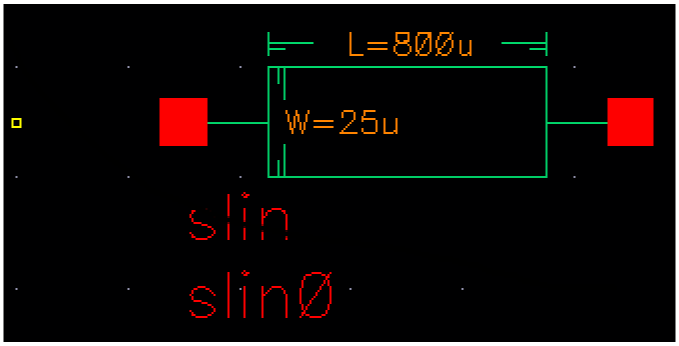
Similar comments apply to coupled microstrip cells like mclin and maclin.

Example

```
mlin0 (net1 net2 0 0) mtline stackup=stackup0 layer=["signalTop"] len=800u
linewidth=[25u]
```


© 1994-2020 All Rights Reserved.

SYMBOL: SLIN



Stripline

Command-line help

```
spectre -h mtline
```

CDF Parameters

Label	Parameter	Type	Units	Default
Stackup Instance	Stackup	Instance name		stackup0
Signal Layer	Layer	String array		strip
Length	Len	Double	meter	800u
Width	Linewidth	Double array	meter	25u

Note: slin is not limited to traditional stripline configurations in the strictest sense of the word - with the conductor in a homogeneous dielectric equidistant from the ground planes. The stackup can include any number of dielectric layers of arbitrary material properties, and the conductor need not be equidistant from the two ground planes.

The slin cell was provided for convenience and for naming compatibility with legacy microwave EDA tools but is more general than legacy implementations.

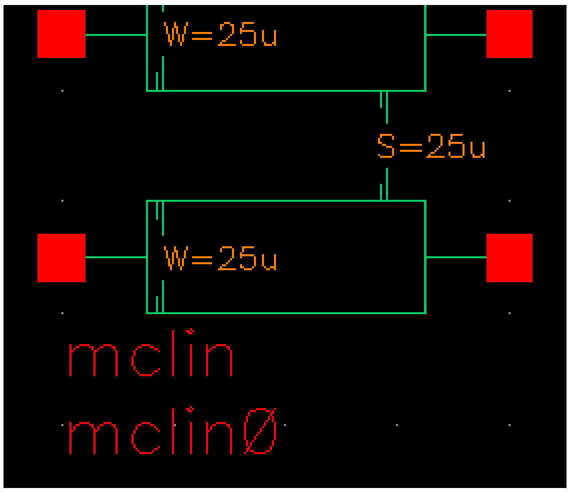
Similar comments apply to coupled stripline cells.

© 1994-2020 All Rights Reserved.

Example

```
slin0 (net1 net2 0 0) mtline stackup=stackup0 layer=["strip"]  len=800u
linewidth=[25u]
```

SYMBOL: MCLIN



Coupled microstrip

Command-line help

```
spectre -h mtline
```

CDF Parameters

Label	**Parameter**	**Type**	**Units**	**Default**
Stackup Instance	Stackup	Instance name		stackup0
Signal Layer	Layer	String array		signalTop
Length	Len	Double	meter	800u
Width	Linewidth	Double array	meter	25u

© 1994-2020 All Rights Reserved.

Spacing	Linespace	Double array	meter	25u

Note: If you need more than 2 coupled conductors, please use the nclin cell.

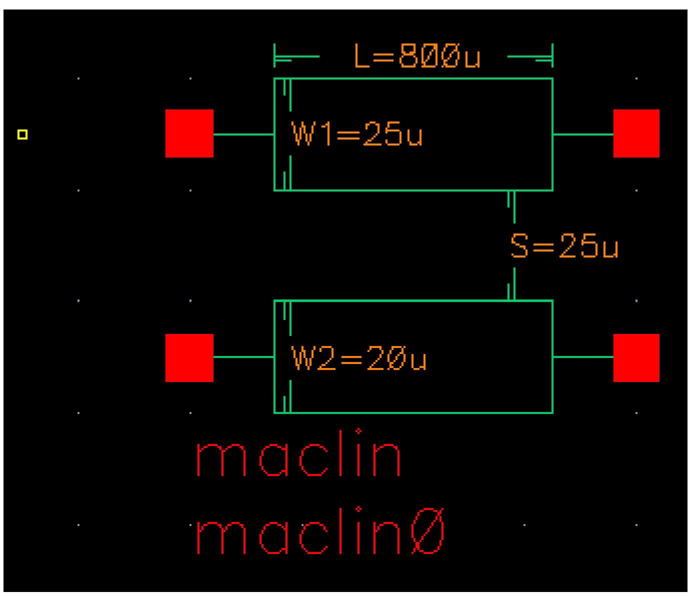
Note: mclin is not limited to traditional microstrip configurations in the strictest sense of the word -- with air above the conductor and a homogeneous dielectric beneath. In fact, the stackup can include any number of dielectric layers of arbitrary material properties, and an optional ground shield above the conductor.

Note: mclin is in effect a general single conductor line above ground (with an optional ground shield). The mlin cell was provided for convenience and for naming compatibility with legacy microwave EDA tools.

Example

```
mclin0 (net1 net2 net3 net4 0 0) mtline stackup=stackup0 layer=["signalTop"]
len=800u linewidth=[25u 25u] linespace=[25u]
```

SYMBOL: MACLIN



Asymmetric coupled microstrip

This cell is similar to `mclin`, but it allows you to specify lines of different width.

© 1994-2020
All Rights Reserved.

Command-line help

```
spectre -h mtline
```

CDF Parameters

Label	Parameter	Type	Units	Default
Stackup Instance	Stackup	Instance name		stackup0
Signal Layer	Layer	String array		signalTop
Length	Len	Double	meter	800u
Width1	Linewidth	Doube array	meter	25u
Width2	Linewidth	Doube array	meter	25u
Spacing	Linespace	Double array	meter	25u

Note: If you need more than 2 coupled conductors, please use the nclin cell.

Note: maclin is not limited to traditional microstrip configurations in the strictest sense of the word -- with air above the conductor and a homogeneous dielectric beneath. In fact, the stackup can include any number of dielectric layers of arbitrary material properties, and an optional ground shield above the conductor.

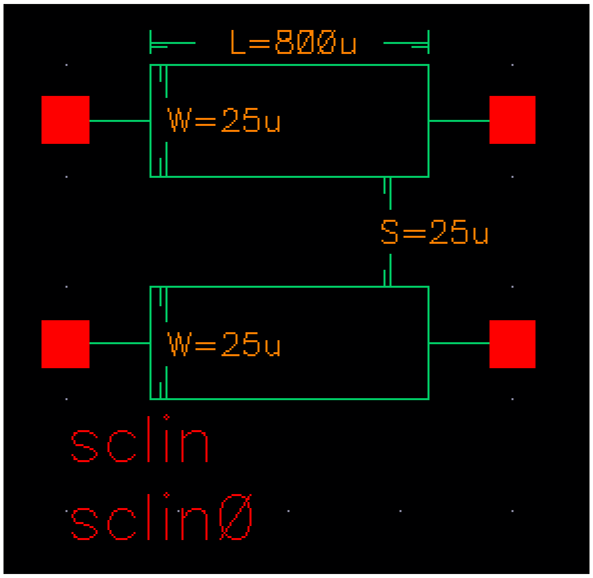
maclin is in effect a general single conductor line above ground (with an optional ground shield). The mlin cell was provided for convenience and for naming compatibility with legacy microwave EDA tools.

Example

```
maclin0 (net1 net2 net3 net4 0 0) mtline stackup=stackup0 layer=["signalTop"]
len=800u linewidth=[25u 25u] linespace=[25u]
```


© 1994-2020 All Rights Reserved.

SYMBOL: SCLIN



Coupled stripline

Command-line help

```
spectre -h mtline
```

CDF Parameters

Label	**Parameter**	**Type**	**Units**	**Default**
Stackup Instance	Stackup	Instance name		stackup0
Signal Layer	Layer	String array		strip
Length	Len	Double	meter	800u
Width	Linewidth	Doube array	meter	25u
Spacing	Linespace	Double array	meter	25u

Note: If you need more than 2 coupled conductors, please use the nclin cell.

© 1994-2020 All Rights Reserved.

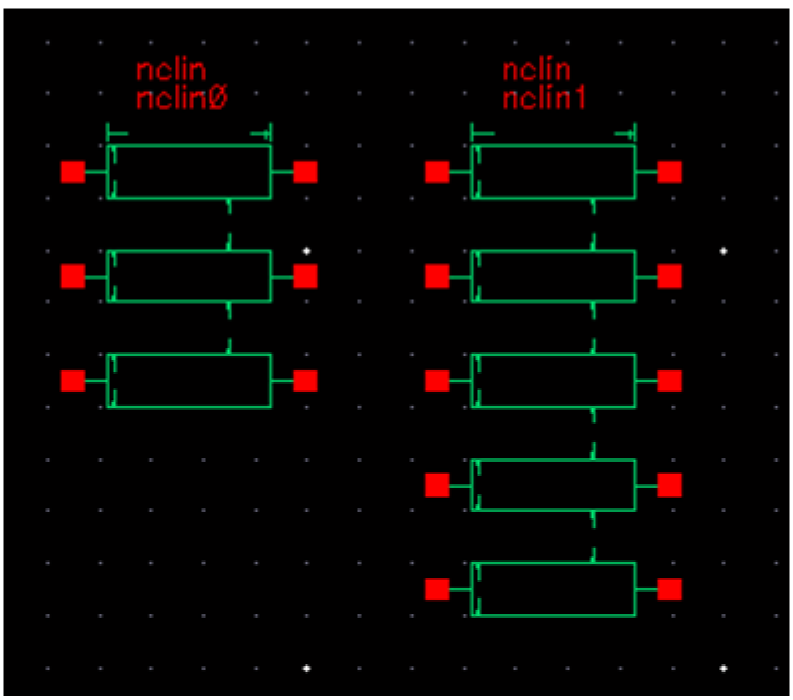
Note: sclin is not limited to traditional stripline configurations in the strictest sense of the word - with the conductor in a homogeneous dielectric equidistant from the ground planes. The stackup can include any number of dielectric layers of arbitrary material properties, and the conductor need not be equidistant from the two ground planes.

The sclin cell was provided for convenience and for naming compatibility with legacy microwave EDA tools but is more general than legacy implementations.

Example

```
sclin0 (net1 net2 net3 net4 0 0) mtline stackup=stackup0 layer=["strip"] len=800u
linewidth=[25u 25u] linespace=[25u]
```

SYMBOL: NCLIN



N coupled conductors

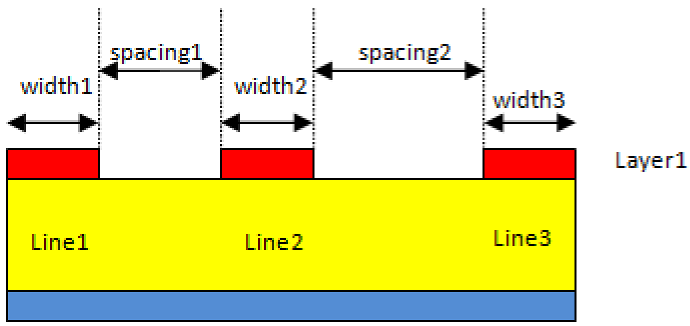
`nclin` is a parametric cell which lets you specify a transmission line consisting of N coupled conductors. N is between 1 and 64.

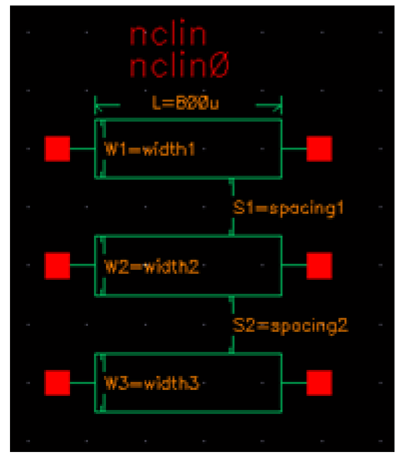
For an N conductor line, you specify N widths and N-1 spacings. In the CDF form, the widths and spacings are specified using parameters `Width1`, `Width2`, ... `WidthN` and `Spacing1`,

© 1994-2020
All Rights Reserved.

`Spacing2`,..... `SpacingN-1`. On the symbol, `Width1` corresponds to the top line, and `WidthN` corresponds to the bottom line.

Line spacing is measured from edge-to-edge, as shown (cross-section view) below.



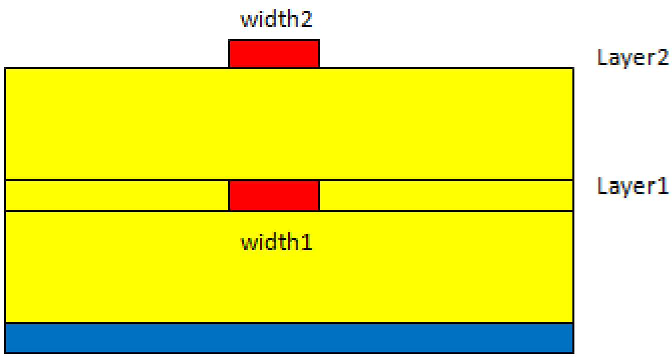


Assuming that a conductor by the name of `Layer1` is defined in the stackup, the CDF form for the above configuration would look, as shown below.

Number of conductor lines	3
Browse and select stackup instance	☐
Stackup instance	stackup0
Browse and select signal layer	☐
Signal layer	layer1
Length	800u M
Width1	width1 M
Width2	width2 M
Width3	width3 M
Spacing1	spacing1 M
Spacing2	spacing2 M

© 1994-2020
All Rights Reserved.

You can also define broad-side coupled lines (coupled lines on multiple layers). Consider the following example:



Since spacing is defined as starting from the right edge of one line to the left edge of the next, we use negative spacing to position them on top of one another. Therefore, the CDF form would look like the following:

Number of conductor lines	2
Browse and select stackup instance	☐
Stackup instance	stackup0
Browse and select signal layer(s)	☐
Signal layer(s)	layer1 layer2
Length	800u M
Width1	width1 M
Width2	width2 M
Spacing1	-width1 M

© 1994-2020
All Rights Reserved.

To specify more than one layer, you can type in a comma-separated list of names, as shown, or use <SHIFT>+click or <CTRL>+click key combinations to select multiple layer names in the *Select signal layer* browser.

Command-line help

```
spectre -h mtline
```

CDF Parameters

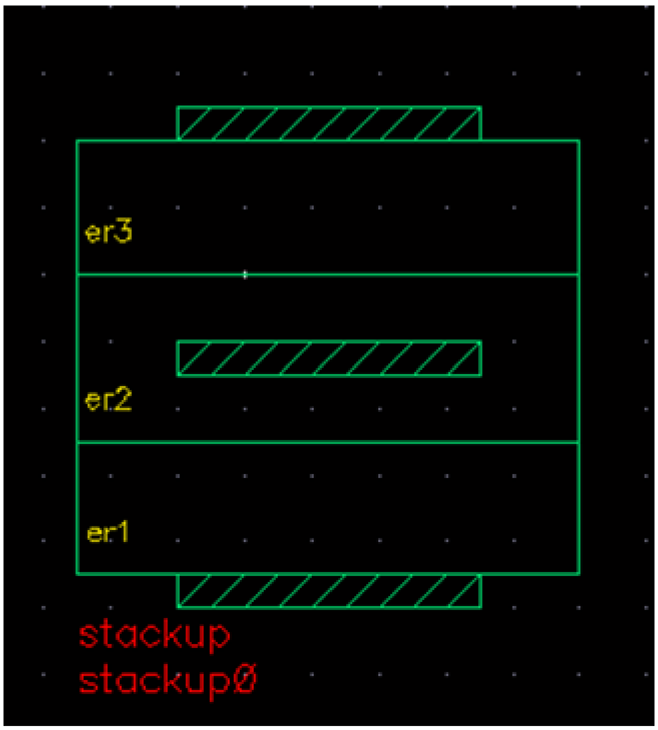
Label	**Parameter**	**Type**	**Units**	**Default**
Number of Conductor Lines	N/A	Integer		1
Stackup Instance	Stackup	Instance Name		stackup0
Signal Layer	Layer	String array		strip
Length	Len	Double	meter	800u
Width	Linewidth	Doube array	meter	25u
Spacing	Linespace	Double array	meter	25u

© 1994-2020 All Rights Reserved.

Example

```
nclin0 (net1 net2 net3 net4 0 0) mtline stackup=stackup0 layer=["strip"] len=800u
linewidth=[25u 25u] linespace=[25u]
```

SYMBOL: STACKUP



Stackup

This cell is used to define common geometric and material properties of transmission lines. The netlisting of the stackup cell includes several Spectre devices, such as stackup, dielectric, and conductor.

Command-line help

```
spectre -h stackup
spectre -h dielectric
spectre -h conductor
```


© 1994-2020
All Rights Reserved.

CDF Parameters

Note: The CDF parameters in the stackup are read-only. Stackup is designed for use behind a graphical editor, as described in the rfTlineLib documentation. The editor sets the stackup CDF parameters automatically.

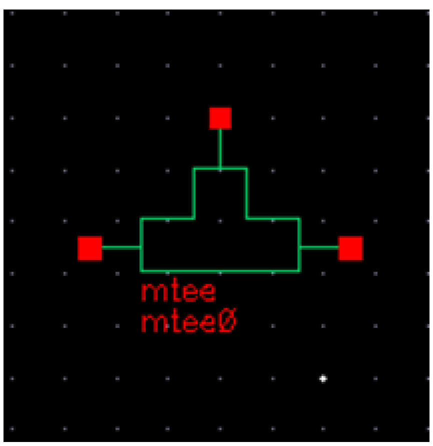
Example

```
stackup0 stackup names=["signalBottom" "dielBottom" "bottom" "core1" "strip"
"sideDiel" "core2" "top" "coreTop" "signalTop" ] type=["Cond" "Diel" "Ref" "Diel"
"Cond" "Diel" "Diel" "Ref" "Diel" "Cond" ] thickness=[35.56u 50.8u 71.12u 25.4u
35.56u 35.56u 25.4u 71.12u 25.4u 35.56u] material=[Copper FR4 Copper FR4  Copper
FR4 FR4  Copper FR4 Copper]
```

```
FR4 dielectric type="wideband_debye" nd=5 data=[1G 4.2 20m 10K 1T]
```

```
Copper conductor type="smooth" nd=1 data=[56M]
```

SYMBOL: MTEE



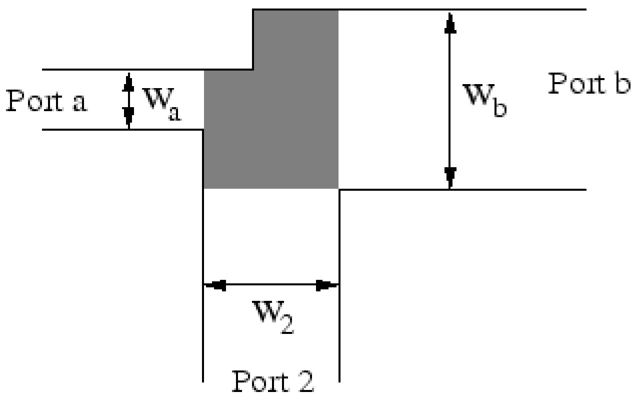
Microstrip tee junction

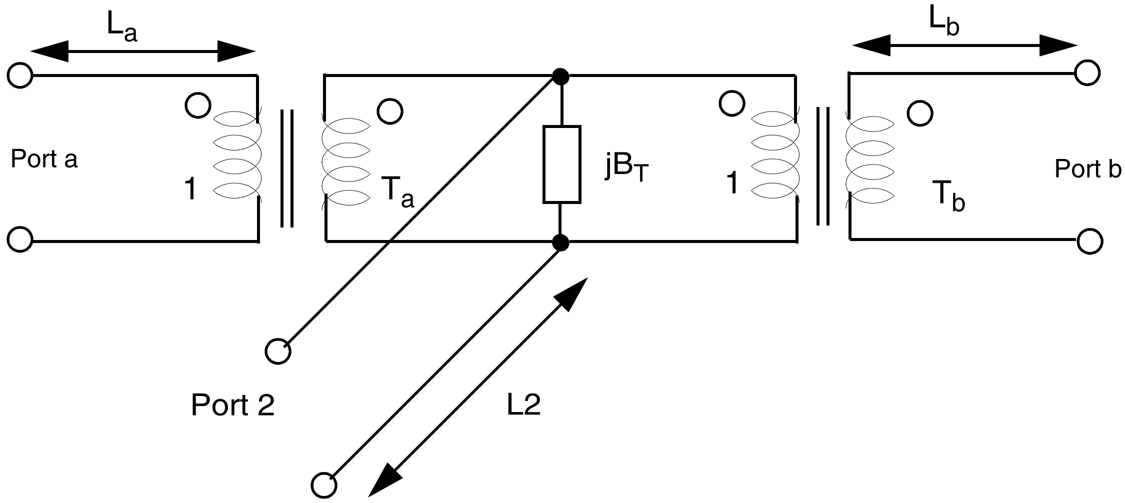
A tee instance must have exactly three transmission line instances connecting to its terminals, one transmission line per terminal. All three transmission lines should refer to the same stackup. It is an error to have instances of any other model connect to the terminals of a tee.

The geometric and material information is obtained automatically from the transmission line instances.

A diagram of the physical microstrip tee junction and the equivalent circuit is shown below.

© 1994-2020
All Rights Reserved.





A model of the microstrip tee junction shows a unsymmetrical microstrip tee with the main arms consisting of port `a` and `b` and with the side arm consisting of port 2. The grayed portion marks the area modeled by the equivalent circuit (shown at the bottom). The equivalent circuit consists of a shunt reactance B_T, one transformer in each main arm (ratios T_a and T_b) and a microstrip line in each arm (width W_a, W_b, and W_2).

The equivalent parallel plate line width is given by:

$$D = \frac{Z_{F0}}{\sqrt{\varepsilon_{r,\,eff}}} \cdot \frac{h}{Z_L}$$

Z_{F0} is the vacuum field impedance, h is the height of substrate, $\varepsilon_{r,\,eff}$ is the effective,relative dielectric constant, and Z_L is the microstrip line impedance.

© 1994-2020
All Rights Reserved.

The first higher order mode cut-off frequency is given by:

$$f_p = 4 \cdot 10^5 \cdot \frac{Z_L}{h}$$

The effective wave length of the microstrip quasi-TEM mode is given by:

$$\lambda = \frac{c_0}{\sqrt{\varepsilon_{r,\,eff}} \cdot f}$$

The main arm displacements of the reference planes from the center lines are (index x stand for a or b) given by:

$$d_x = 0.055 \cdot D_2 \cdot \frac{Z_{L,\,x}}{Z_{L,\,2}} \cdot \left(1 - 2 \cdot \frac{Z_{L,\,x}}{Z_{L,\,2}} \cdot \left(\frac{f}{f_{p,\,x}}\right)^2\right)$$

where f is the frequency.

The length of the line in the main arms is given by:

$$L = 0.5 \cdot W_2 - d_x$$

The side arm displacement of the reference planes from the center lines is given by:

$$d_2 = \sqrt{D_a \cdot D_b} \cdot (0.5 - R \cdot (0.05 + 0.7 \cdot \exp(-1.6 \cdot R) + 0.25 \cdot R \cdot Q - 0.17 \cdot \ln R))$$

The length of the line in the side arm is given by:

$$L_2 = 0.5 \cdot max(W_a, W_b) - d_2$$

where `max (x,y)` is the larger of both the quantities.

© 1994-2020 All Rights Reserved.

R and Q are given by:

$$R = \sqrt{\frac{Z_{L,a} \cdot Z_{L,b}}{Z_{L,2}}} \qquad Q = \frac{f^2}{f_{p,a} \cdot f_{p,b}}$$

The turn ratio of the transformers in the side arms is given by:

$$T_x^{\ 2} = 1 - \pi \cdot \left(\frac{f}{f_{p,x}}\right)^2 \cdot \left(\frac{1}{12} \cdot \left(\frac{Z_{L,x}}{Z_{L,2}}\right)^2 + \left(0.5 - \frac{d_2}{D_x}\right)^2\right)$$

The shunt susceptance is given by:

$$B_T = 5.5 \cdot \sqrt{\frac{D_a \cdot D_b}{\lambda_a \cdot \lambda_b}} \cdot \frac{\varepsilon_r + 2}{\varepsilon_r} \cdot \frac{1}{Z_{L,2} \cdot T_a \cdot T_b} \cdot \frac{\sqrt{d_a \cdot d_b}}{D_2} \cdot \left(1 + 0.9 \cdot \ln R + 4.5 \cdot R \cdot Q - 4.4 \cdot \exp(-1.3 \cdot R) - 20 \cdot \left(\frac{Z_{L,2}}{Z_{F0}}\right)^2\right)$$

© 1994-2020
All Rights Reserved.

For better implementation of the microstrip tee, the device parameter of the internal equivalent circuit (two transformers and the shunt susceptance) are given below. The port numbering for them is port a = 1, port b = 2, and port 2 = 3.

$$(\underline{Y}) = infinity$$

$$(\underline{Z}) = \frac{1}{j \cdot B_T} \cdot \begin{bmatrix} \frac{1}{n_a^2} & \frac{1}{n_a \cdot n_b} & \frac{1}{n_a} \\ \frac{1}{n_a \cdot n_b} & \frac{1}{n_b^2} & \frac{1}{n_b} \\ \frac{1}{n_a} & \frac{1}{n_b} & 1 \end{bmatrix}$$

$$\underline{S}_{11} = \frac{1 - n_a^2 \cdot \left(j \cdot B_T \cdot Z_0 + \frac{1}{n_b^2} + 1\right)}{1 + n_a^2 \cdot \left(j \cdot B_T \cdot Z_0 + \frac{1}{n_b^2} + 1\right)}$$

$$\underline{S}_{22} = \frac{1 - n_b^2 \cdot \left(j \cdot B_T \cdot Z_0 + \frac{1}{n_a^2} + 1\right)}{1 + n_b^2 \cdot \left(j \cdot B_T \cdot Z_0 + \frac{1}{n_a^2} + 1\right)}$$

© 1994-2020 All Rights Reserved.

$$\underline{S}_{33} = \frac{1 - \left(\frac{1}{n_a^{\ 2}} + \frac{1}{n_b^{\ 2}} + j \cdot B_T \cdot Z_0\right)}{1 + \left(\frac{1}{n_a^{\ 2}} + \frac{1}{n_b^{\ 2}} + j \cdot B_T \cdot Z_0\right)}$$

$$\underline{S}_{13} = \underline{S}_{31} = \frac{2 \cdot n_a}{n_a^{\ 2} \cdot \left(\frac{1}{n_b^{\ 2}} + j \cdot B_T \cdot Z_0 + 1\right) + 1}$$

$$\underline{S}_{23} = \underline{S}_{32} = \frac{2 \cdot n_b}{n_b^{\ 2} \cdot \left(\frac{1}{n_a^{\ 2}} + j \cdot B_T \cdot Z_0 + 1\right) + 1}$$

$$\underline{S}_{12} = \underline{S}_{21} = \frac{2}{n_a \cdot n_b \cdot (j \cdot B_T \cdot Z_0 + 1) + \frac{n_a}{n_b} + \frac{n_b}{n_a}}$$

The MNA matrix representation can be derived from the Z parameters in the following way:

$$\begin{bmatrix} \ldots & \ldots & \ldots & 1 & 0 & 0 \\ \ldots & \ldots & \ldots & 0 & 1 & 0 \\ \ldots & \ldots & \ldots & 0 & 0 & 1 \\ -1 & 0 & 0 & Z_{11} & Z_{12} & Z_{13} \\ 0 & -1 & 0 & Z_{21} & Z_{22} & Z_{23} \\ 0 & 0 & -1 & Z_{31} & Z_{32} & Z_{33} \end{bmatrix} \cdot \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ I_{1,\,in} \\ I_{2,\,in} \\ I_{3,\,in} \end{bmatrix} = \begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

© 1994-2020 All Rights Reserved.

Note that the main arm displacements yield two small microstrip lines at each main arm and the side arm displacement results in a small microstrip strip line as well, but with negative length, that is a kind of phase shifter here.

The transformer ratios are negative with increasing frequency which produces complex values in the Z-parameter matrix as well as in the S-parameter matrix. That is why the ratios are delimited to a minimum value.

Spectre uses an augmented version of the Hammerstad model by accounting for the frequency-dependent characteristic impedance and the effective dielectric constant of the connecting transmission lines.

Command-line help

```
spectre -h tee
```

CDF Parameters

`mtee` has no CDF parameters

Range of Usage

The T-junction model is valid for the frequency range not exceeding the first high order mode cutoff frequency. The first high order model cutoff frequency, fp, can be calculated using

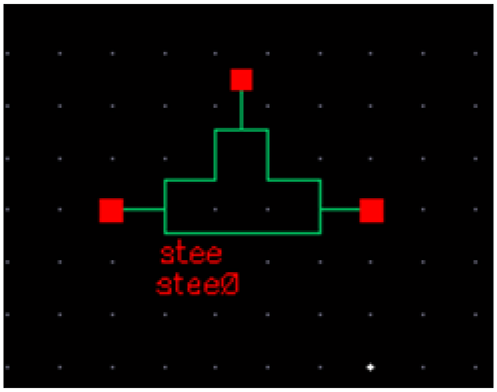
fp [GHz]= Zo/h[mm], where Zo is the characteristic impedance of the trace, and h is the substrate thickness. The results of the model for frequencies above the fp are not correct. The model achieves the best accuracy when the ratio of the trace width Wi to the substrate thickness h satisfies the inequality 1<=Wi/h <=10. The dielectric constant of the substrate should not exceed 12.8.

Example

```
mtee0 (net1 net2 net3) tee
```


© 1994-2020 All Rights Reserved.

SYMBOL: STEE



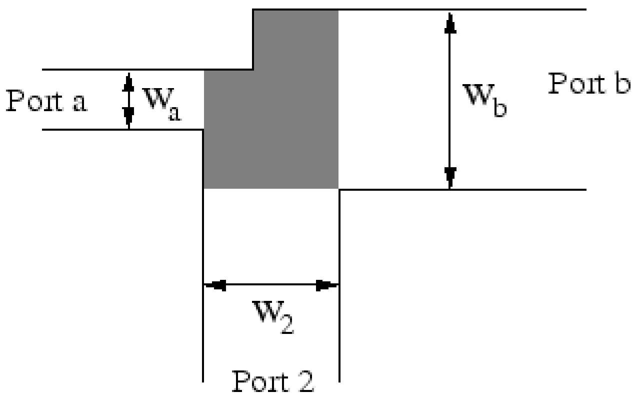
Stripline tee junction

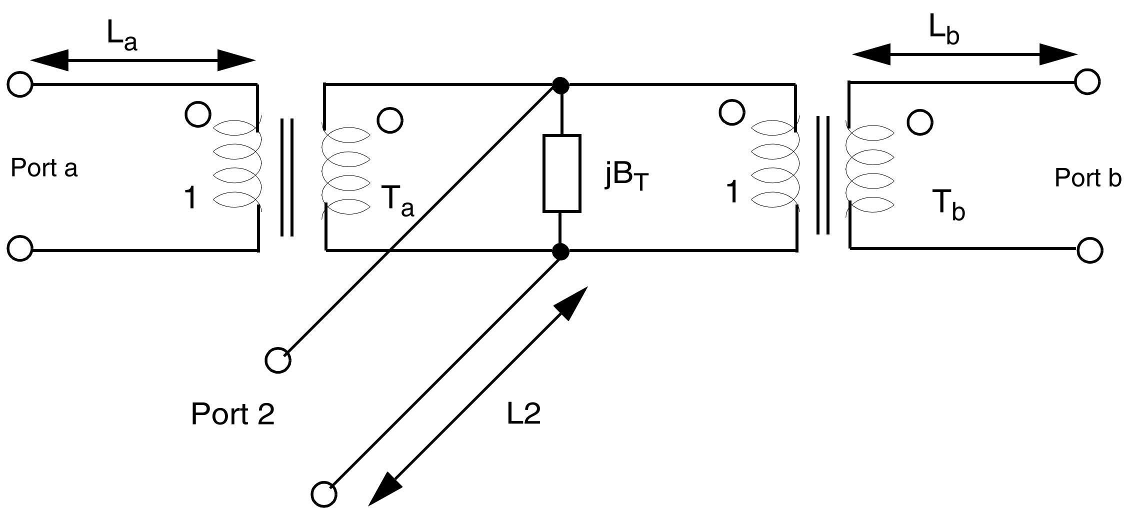
A tee instance must have exactly three transmission line instances connecting to its terminals, one transmission line per terminal. All three transmission lines should refer to the same stackup. It is an error to have instances of any other model connect to the terminals of a tee.

The geometric and material information is obtained automatically from the transmission line instances.

A diagram of the physical microstrip tee junction and the equivalent circuit is shown below.

© 1994-2020 All Rights Reserved.





A model of the microstrip tee junction shows a unsymmetrical microstrip tee with the main arms consisting of port `a` and `b` and with the side arm consisting of port 2. The grayed portion marks the area modeled by the equivalent circuit (shown at the bottom). The equivalent circuit consists of a shunt reactance B_T, one transformer in each main arm (ratios T_a and T_b) and a microstrip line in each arm (width W_a, W_b, and W_2).

The equivalent parallel plate line width is given by:

$$D = \frac{Z_{F0}}{\sqrt{\varepsilon_{r,\,eff}}} \cdot \frac{h}{Z_L}$$

Z_{F0} is the vacuum field impedance, h is the height of substrate, $\varepsilon_{r,\,eff}$ is the effective, relative dielectric constant, and Z_L is the microstrip line impedance.

© 1994-2020 All Rights Reserved.

The first higher order mode cut-off frequency is given by:

$$f_p = 4 \cdot 10^5 \cdot \frac{Z_L}{h}$$

The effective wave length of the microstrip quasi-TEM mode is given by:

$$\lambda = \frac{c_0}{\sqrt{\varepsilon_{r,\,eff}} \cdot f}$$

The main arm displacements of the reference planes from the center lines are (index x stand for a or b) given by:

$$d_x = 0.055 \cdot D_2 \cdot \frac{Z_{L,\,x}}{Z_{L,\,2}} \cdot \left(1 - 2 \cdot \frac{Z_{L,\,x}}{Z_{L,\,2}} \cdot \left(\frac{f}{f_{p,\,x}}\right)^2\right)$$

where f is the frequency.

The length of the line in the main arms is given by:

$$L = 0.5 \cdot W_2 - d_x$$

The side arm displacement of the reference planes from the center lines is given by:

$$d_2 = \sqrt{D_a \cdot D_b} \cdot (0.5 - R \cdot (0.05 + 0.7 \cdot \exp(-1.6 \cdot R) + 0.25 \cdot R \cdot Q - 0.17 \cdot \ln R))$$

The length of the line in the side arm is given by:

$$L_2 = 0.5 \cdot max(W_a, W_b) - d_2$$

where `max (x,y)` is the larger of both the quantities.

© 1994-2020 All Rights Reserved.

R and Q are given by:

$$R = \sqrt{\frac{Z_{L,a} \cdot Z_{L,b}}{Z_{L,2}}} \qquad Q = \frac{f^2}{f_{p,a} \cdot f_{p,b}}$$

The turn ratio of the transformers in the side arms is given by:

$$T_x^{\,2} = 1 - \pi \cdot \left(\frac{f}{f_{p,x}}\right)^2 \cdot \left(\frac{1}{12} \cdot \left(\frac{Z_{L,x}}{Z_{L,2}}\right)^2 + \left(0.5 - \frac{d_2}{D_x}\right)^2\right)$$

The shunt susceptance is given by:

$$B_T = 5.5 \cdot \sqrt{\frac{D_a \cdot D_b}{\lambda_a \cdot \lambda_b}} \cdot \frac{\varepsilon_r + 2}{\varepsilon_r} \cdot \frac{1}{Z_{L,2} \cdot T_a \cdot T_b} \cdot \frac{\sqrt{d_a \cdot d_b}}{D_2} \cdot \left(1 + 0.9 \cdot \ln R + 4.5 \cdot R \cdot Q - 4.4 \cdot \exp(-1.3 \cdot R) - 20 \cdot \left(\frac{Z_{L,2}}{Z_{F0}}\right)^2\right)$$

© 1994-2020 All Rights Reserved.

For better implementation of the microstrip tee, the device parameter of the internal equivalent circuit (two transformers and the shunt susceptance) are given below. The port numbering for them is port a = 1, port b = 2, and port 2 = 3.

$$(\underline{Y}) = infinity$$

$$(\underline{Z}) = \frac{1}{j \cdot B_T} \cdot \begin{bmatrix} \frac{1}{n_a^2} & \frac{1}{n_a \cdot n_b} & \frac{1}{n_a} \\ \frac{1}{n_a \cdot n_b} & \frac{1}{n_b^2} & \frac{1}{n_b} \\ \frac{1}{n_a} & \frac{1}{n_b} & 1 \end{bmatrix}$$

$$\underline{S}_{11} = \frac{1 - n_a^2 \cdot \left(j \cdot B_T \cdot Z_0 + \frac{1}{n_b^2} + 1\right)}{1 + n_a^2 \cdot \left(j \cdot B_T \cdot Z_0 + \frac{1}{n_b^2} + 1\right)}$$

$$\underline{S}_{22} = \frac{1 - n_b^2 \cdot \left(j \cdot B_T \cdot Z_0 + \frac{1}{n_a^2} + 1\right)}{1 + n_b^2 \cdot \left(j \cdot B_T \cdot Z_0 + \frac{1}{n_a^2} + 1\right)}$$

© 1994-2020 All Rights Reserved.

$$\underline{S}_{33} = \frac{1 - \left(\frac{1}{n_a^{\;2}} + \frac{1}{n_b^{\;2}} + j \cdot B_T \cdot Z_0\right)}{1 + \left(\frac{1}{n_a^{\;2}} + \frac{1}{n_b^{\;2}} + j \cdot B_T \cdot Z_0\right)}$$

$$\underline{S}_{13} = \underline{S}_{31} = \frac{2 \cdot n_a}{n_a^{\;2} \cdot \left(\frac{1}{n_b^{\;2}} + j \cdot B_T \cdot Z_0 + 1\right) + 1}$$

$$\underline{S}_{23} = \underline{S}_{32} = \frac{2 \cdot n_b}{n_b^{\;2} \cdot \left(\frac{1}{n_a^{\;2}} + j \cdot B_T \cdot Z_0 + 1\right) + 1}$$

$$\underline{S}_{12} = \underline{S}_{21} = \frac{2}{n_a \cdot n_b \cdot (j \cdot B_T \cdot Z_0 + 1) + \frac{n_a}{n_b} + \frac{n_b}{n_a}}$$

The MNA matrix representation can be derived from the Z parameters in the following way:

$$\begin{bmatrix} \dots & \dots & \dots & 1 & 0 & 0 \\ \dots & \dots & \dots & 0 & 1 & 0 \\ \dots & \dots & \dots & 0 & 0 & 1 \\ -1 & 0 & 0 & Z_{11} & Z_{12} & Z_{13} \\ 0 & -1 & 0 & Z_{21} & Z_{22} & Z_{23} \\ 0 & 0 & -1 & Z_{31} & Z_{32} & Z_{33} \end{bmatrix} \cdot \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ I_{1,\,in} \\ I_{2,\,in} \\ I_{3,\,in} \end{bmatrix} = \begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

© 1994-2020 All Rights Reserved.

Note that the main arm displacements yield two small microstrip lines at each main arm and the side arm displacement results in a small microstrip strip line as well, but with negative length, that is a kind of phase shifter here.

The transformer ratios are negative with increasing frequency which produces complex values in the Z-parameter matrix as well as in the S-parameter matrix. That is why the ratios are delimited to a minimum value.

Spectre uses an augmented version of this model by accounting for the frequency-dependent characteristic impedance and the effective dielectric constant of the connecting transmission lines.

Command-line help

```
spectre -h tee
```

CDF Parameters

`stee` has no CDF parameters

Range of Usage

The model achieves the best accuracy for W1=W2, where W1 and W2 are the trace with for the traces connecting to the "top of the T". In case of $W1 \neq W2$, the geometric mean `sqrt(W1*W2)` is used which works for the trace width mismatch of up to 10%.

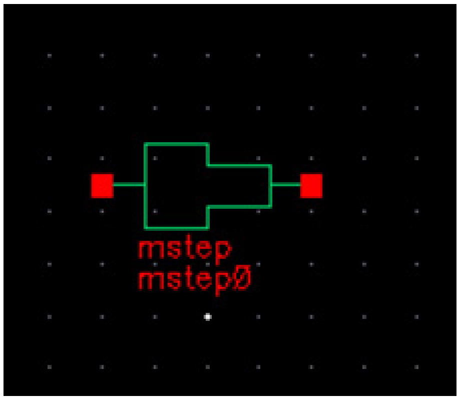
Denote the characteristic impedances of the transmission lines of the width W1 and W2 (connecting to the "top of the T") as Zo1 and that of the width W3 as Zo3. The condition `0.2 <=Zo1/Zo3 <=2` is required for the maximum accuracy. The stripline T-junction model is accurate for the frequency range up to the first high-order cut-off frequency.

Example

```
stee0 (net1 net2 net3) tee
```


© 1994-2020 All Rights Reserved.

SYMBOL: MSTEP



Microstrip step in width (centered)

A step instance must have exactly two transmission line instances connecting to its terminals, one transmission line per terminal. Both transmission lines should refer to the same stackup. It is an error to have instances of any other model connect to the terminals of a step.

The geometric and material information is obtained automatically from the transmission line instances. The step junction model is based on the approximate full-wave EM analysis that involves transforming to an equivalent parallel plate waveguide and the use of mode matching technique, as described in:

T.S. Chu, T. Itoh, and Y.C. Shih 'Comparative Study of Mode-Matching Formulations for Microstrip Discontinuity Problems', IEEE Trans. MTT, V. 33, N. 10, 1985,and references within.

The advantage of this approach over the closed-form empirical formulas is its validity for a wide range of the dielectric constants.

Command-line help

```
spectre -h step
```

CDF Parameters

`mstep` has no CDF parameters

© 1994-2020 All Rights Reserved.

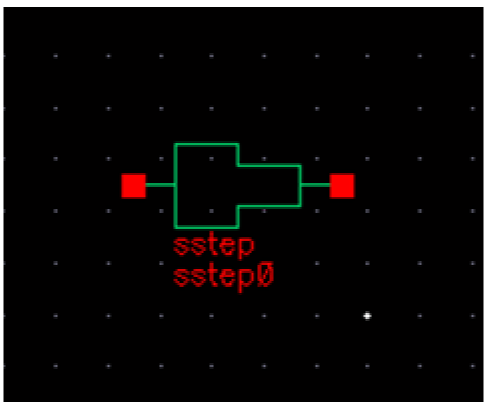
Range of Usage

The model is valid for the relative dielectric constant ranging from 1 to 12.8 (even though there is no hard upper limit), and the ratio of the trace width from 1 (for which the model acts as a short) to 20.

Example

```
mstep0 (net1 net2) step
```

SYMBOL: SSTEP



Stripline step in width (centered)

A step instance must have exactly two transmission line instances connecting to its terminals, one transmission line per terminal. Both transmission lines should refer to the same stackup. It is an error to have instances of any other model connect to the terminals of a step.

The geometric and material information is obtained automatically from the transmission line instances. The step junction model is based on the approximate full-wave EM analysis that involves transforming to an equivalent parallel plate waveguide and the use of mode matching technique, as described in:

T.S. Chu, T. Itoh, and Y.C. Shih 'Comparative Study of Mode-Matching Formulations for Microstrip Discontinuity Problems', IEEE Trans. MTT, V. 33, N. 10, 1985, and references within.

The advantage of this approach over the closed-form empirical formulas is its validity for a wide range of the dielectric constants.

© 1994-2020 All Rights Reserved.

Command-line help

```
spectre -h step
```

CDF Parameters

`sstep` has no CDF parameters

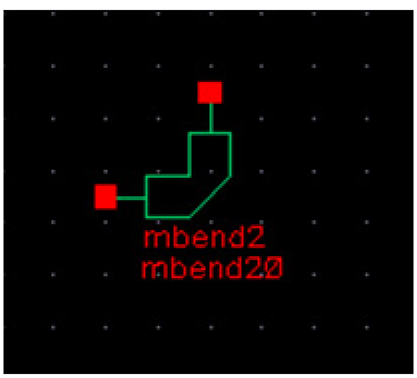
Range of Usage

The model is valid for the relative dielectric constant ranging from 1 to 12.8 (even though there is no hard upper limit), and the ratio of the trace width from 1 (in which case the model acts as a short) to 20.

Example

```
sstep0 (net1 net2) step
```

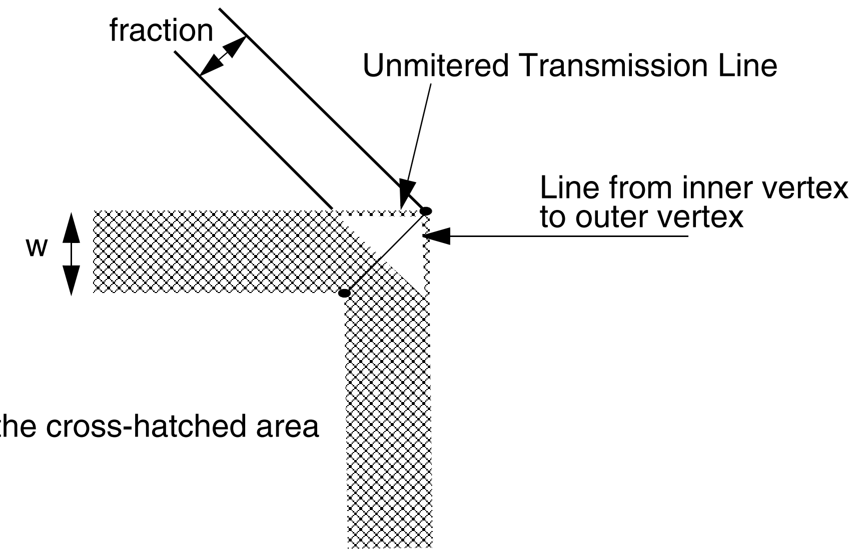
SYMBOL: MBEND2



© 1994-2020 All Rights Reserved.

Microstrip optimally mitered 90 degree bend

The geometry of the MBEND2 model is defined in the figure below. Both traces have the same width, W. The miter fraction M is defined as *M=fraction/Line from inner to outer vertex*, where fraction and line from inner to outer vertex are defined below.



The optimal miter fraction was determined experimentally [R. J. P. Douville and D. S. James, Experimental study of symmetric microstrip bends and their compensation; IEEE Trans. Microwave Theory Tech., vol. MTT-26, pp. 175-182, Mar. 1978] to be well approximated as `M=(0.52+0.65exp[-1.35w/h])`.

A `bend2` instance must have exactly two transmission line instances connecting to its terminals, one transmission line per terminal. Both transmission lines should refer to the same stackup. It is an error to have instances of any other model connect to the terminals of a step.

The geometric and material information is obtained automatically from the transmission line instances.

The frequency-domain model is an empirically based model, which consists of a lumped equivalent circuit (T-network of two inductors and a shunt capacitor to ground). The equivalent circuit parameters are frequency-independent and are expressed as functions of trace width, dielectric height, and dielectric constant.

The expressions were developed by *Kirschning*, *Jansen*, and *Koster* and given in the following publication:

© 1994-2020
All Rights Reserved.

M. Kirschning, R. H. Jansen, and N. H. L. Koster. 'Measurement and Computer-Aided Modeling of Microstrip Discontinuities by an Improved Resonator Method', 1983 IEEE MTT-S International Microwave Symposium Digest, May 1983, pp. 495-497.

Command-line help

```
spectre -h mbend2
```

CDF Parameters

`mbend2` has no CDF parameters

Range of Usage

The ratio of the trace width W to the substrate thickness h should satisfy 0.2<= W/h<=6.0. For the best accuracy, the dielectric constant Er should satisfy

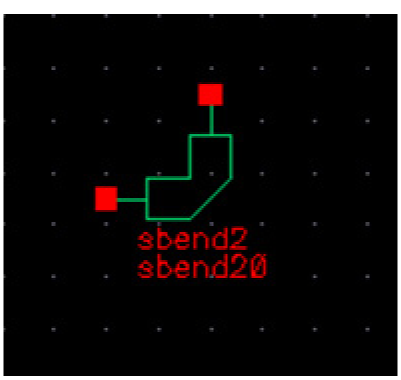
2.36 <= Er <= 10.4

The simulation frequency f should satisfy f[GHz]<12/h[mm].

Example

```
mbend20 (net1 net2) bend2
```

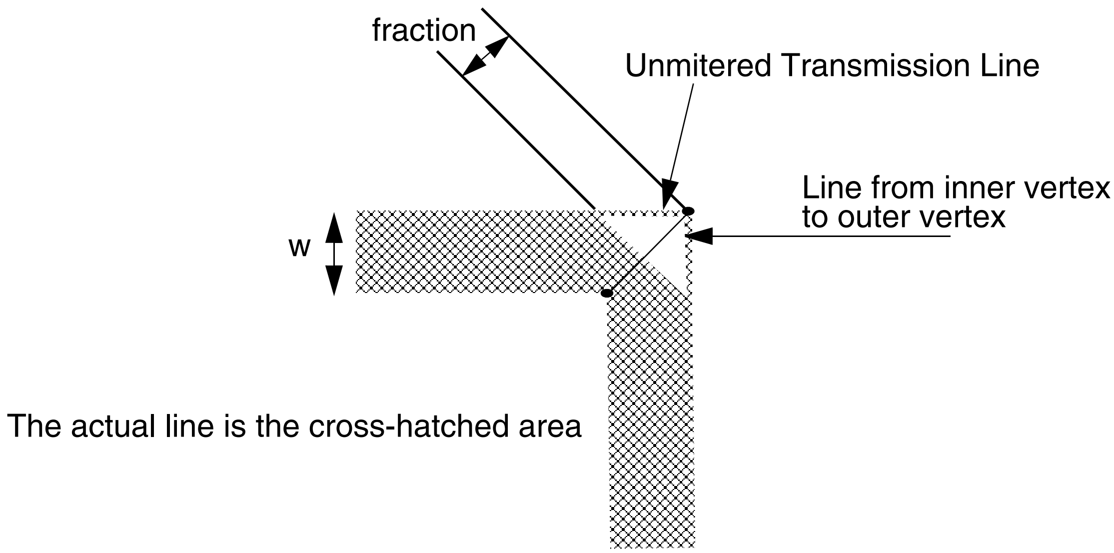
SYMBOL: SBEND2



© 1994-2020 All Rights Reserved.

Stripline optimally mitered 90 degree bend

The geometry of the MBEND2 model is defined in the figure below. Both traces have the same width, W. The miter fraction M is defined as *M=fraction/Line from inner to outer vertex*, where fraction and the line are defined below.



A bend2 instance must have exactly two transmission line instances connecting to its terminals, one transmission line per terminal. Both transmission lines should refer to the same stackup. It is an error to have instances of any other model connect to the terminals of a step.

The geometric and material information is obtained automatically from the transmission line instances.

The frequency-domain model is an empirically based model which consists of a lumped equivalent circuit (T-network of two inductors and a shunt capacitor to ground). The equivalent circuit parameters are frequency-independent and are expressed as functions of trace width, dielectric height, and dielectric constant.

The expressions were developed by *Kirschning*, *Jansen*, and *Koster* and given in the following publication:

M. Kirschning, R. H. Jansen, and N. H. L. Koster. 'Measurement and Computer-Aided Modeling of Microstrip Discontinuities by an Improved Resonator Method', 1983 IEEE MTT-S International Microwave Symposium Digest, May 1983, pp. 495-497.

© 1994-2020
All Rights Reserved.

Command-line help

```
spectre -h bend2
```

CDF Parameters

`sbend2` has no CDF parameters

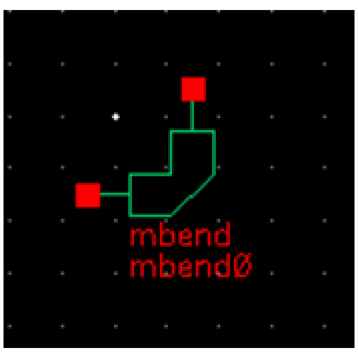
Range of Usage

The ratio of the trace width W to the total dielectric thickness B should satisfy
$0.2 \le w/B \le 3$

Example

```
sbend20 (net1 net2) sbend2
```

SYMBOL: MBEND



Microstrip bend with the specified angle and miter fraction

A bend instance must have exactly two transmission line instances connecting to its terminals, one transmission line per terminal. Both transmission lines should refer to the same stackup. It is an error to have instances of any other model connect to the terminals of a step.

The geometric and material information is obtained automatically from the transmission line instances.

The frequency-domain model is an empirically based model, which consists of a T-network of three impedances with two cascaded sections of transmission lines.

© 1994-2020 All Rights Reserved.

Command-line help

```
spectre -h bend
```

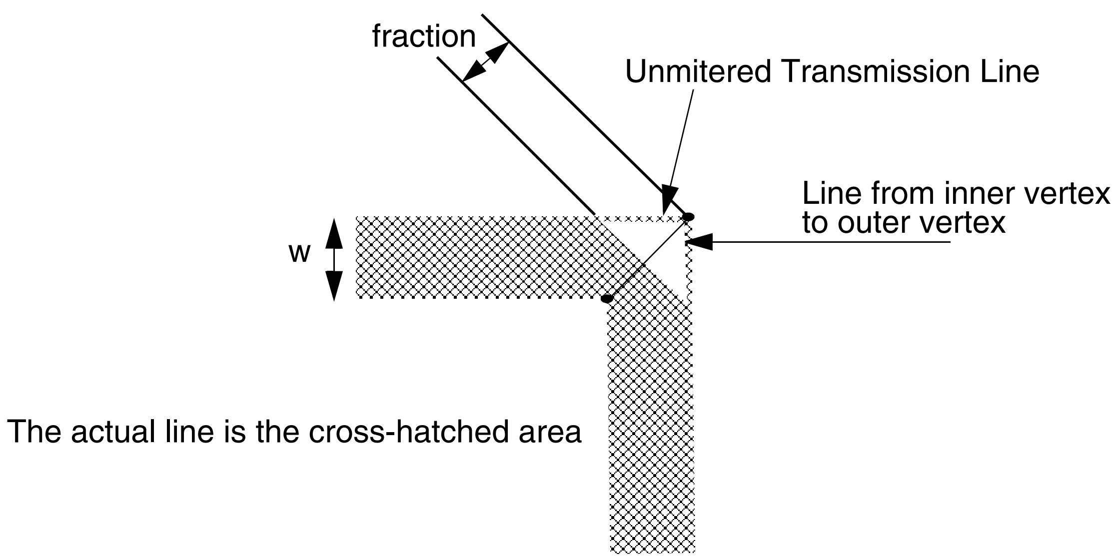
CDF Parameters

Label	**Parameter**	**Type**	**Units**	**Default**
Angle of bend (degrees)	Angle	Double	Degrees	90
Miter Fraction	miterfraction	Double	N/A	0.68805

The miter angle is half the angle of the bend.

The miter fraction is defined as follows:

Draw a line from the inside vertex of the bend to the outside vertex of the unmitered bend. The miter fraction is the distance from the mitered edge at the intersection of the miter and the line to the outside vertex (labeled `fraction` below) divided by the total length of the line from vertex to vertex.



Range of Usage

The dielectric constant Er should satisfy the inequality $1 \le Er \le 12.8$.

The angle parameter (in degrees) should satisfy $-90 \le Angle \le 90$.

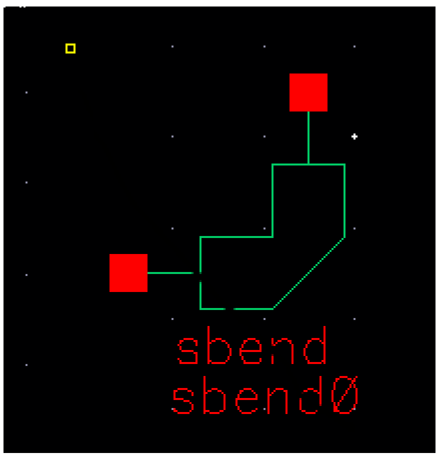
© 1994-2020
All Rights Reserved.

The ration of the trace width W to the substrate thickness h should satisfy the equation $0.01 \le W/h \le 100$.

Example

```
mbend0 (net1 net2) bend angle=85 miterfraction=0.6
```

SYMBOL: SBEND



Microstrip bend with the specified angle and miter fraction

A bend instance must have exactly two transmission line instances connecting to its terminals, one transmission line per terminal. Both transmission lines should refer to the same stackup. It is an error to have instances of any other model connect to the terminals of a step.

The geometric and material information is obtained automatically from the transmission line instances.

The frequency-domain model is an empirically based model, which consists of a T-network of three impedances with two cascaded sections of transmission lines.

Command-line help

```
spectre -h bend
```

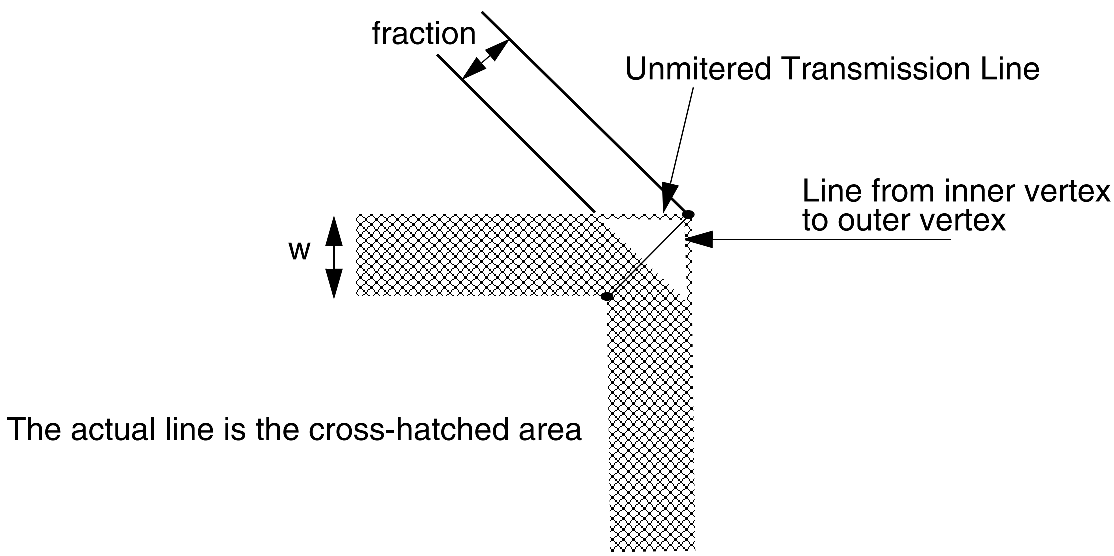

© 1994-2020 All Rights Reserved.

CDF Parameters

Label	Parameter	Type	Units	Default
Angle of bend (degrees)	Angle	Double	Degrees	90
Miter Fraction	miterfraction	Double	N/A	0.68805

The miter fraction is defined as follows:

Draw a line from the inside vertex of the bend to the outside vertex of the unmitered bend. The miter fraction is the distance from the mitered edge at the intersection of the miter and the line to the outside vertex (labeled fraction below) divided by the total length of the line from vertex to vertex.



Range of Usage

The dielectric constant Er should satisfy the inequalities $1 \le Er \le 12.8$.

The ratio of the trace width W to the dielectric substrate thickness h should satisfy the inequality .$0.01 \le W/h \le 100$

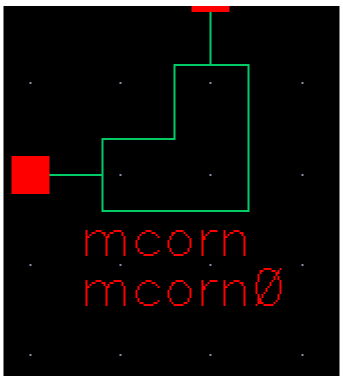
The angle (in degrees) should satisfy the condition $20 \le Angle \le 150$.

© 1994-2020
All Rights Reserved.

Example

```
mbend0 (net1 net2) bend angle=85 miterfraction=0.6
```

SYMBOL: MCORN



Microstrip unmitered 90 degree corner

A corner instance must have exactly two transmission line instances connecting to its terminals, one transmission line per terminal. Both transmission lines should refer to the same stackup. It is an error to have instances of any other model connect to the terminals of a step.

The geometric and material information is obtained automatically from the transmission line instances.

The frequency-domain model is an empirically based model which consists of a lumped equivalent circuit (T-network of two inductors and a shunt capacitor to ground). The equivalent circuit parameters are frequency-independent and are expressed as functions of trace width, dielectric height, and dielectric constant.

The expressions were developed by *Kirschning*, *Jansen*, and *Koster* and given in the following publication:

M. Kirschning, R. H. Jansen, and N. H. L. Koster. 'Measurement and Computer-Aided Modeling of Microstrip Discontinuities by an Improved Resonator Method', 1983 IEEE MTT-S International Microwave Symposium Digest, May 1983, pp. 495-497.

Command-line help

```
spectre -h corner
```


© 1994-2020
All Rights Reserved.

CDF Parameters

`mcorn` has no CDF parameters

Range of Usage

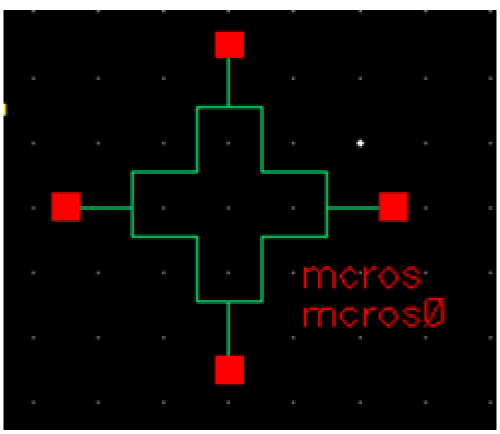
The model, implemented as described in the reference given above, is valid for the ratio of the trace width W to the substrate thickness h within the range $0.2 \le W/h \le 6.0$.

The dielectric constant Er should satisfy the inequality $2.36 \le Er \le 10.4$.

Example

```
mcorn0 (net1 net2) mcorn
```

SYMBOL: MCROS



Microstrip cross

A cross instance must have exactly four transmission line instances connecting to its terminals, one transmission line per terminal. All four transmission lines should refer to the same stackup. It is an error to have instances of any other model connect to the terminals of a step.

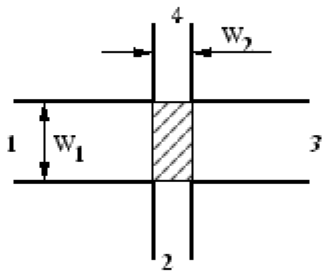
The geometric and material information is obtained automatically from the transmission line instances.

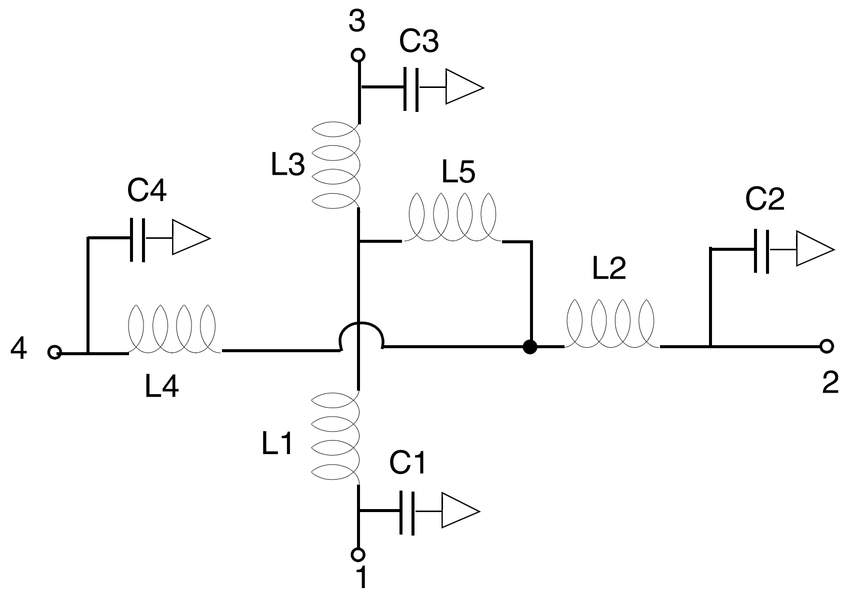
The frequency-domain model is an empirically based model which consists of a lumped equivalent circuit (five inductors and four shunt capacitors to ground). The equivalent circuit

© 1994-2020
All Rights Reserved.

parameters are frequency-independent and are expressed as functions of trace width, dielectric thickness, and dielectric constant.

A diagram of the physical cross and the equivalent circuit is shown below.





The model of a microstrip cross instance shows the equivalent circuit (right-hand side) and the layout with dimensions (left-hand side). The hatched area in the layout marks the area modeled by the equivalent circuit. As can be seen, the model requires the microstrip width of line 1 and 3, as well as the one of line 2 and 4 to be equal to each other. In addition, the

© 1994-2020
All Rights Reserved.

permittivity of the substrate must be ε_r = 9.9 (Accommodation of different dielectric constants is shown later). The component values are calculated as follows:

$$X = \log 10\left(\frac{W_1}{h}\right) \cdot \left(86.6 \cdot \frac{W_2}{h} - 30.9 \cdot \sqrt{\frac{W_2}{h}} + 367\right) + \left(\frac{W_2}{h}\right)^3 + 74 \cdot \frac{W_2}{h} + 130$$

$$C_1 = C_2 = C_3 = C_4$$

$$= 10^{-12} W_1 \cdot \left(0.25 \cdot X \cdot \left(\frac{h}{W_1}\right)^{\frac{1}{3}} - 60 + \frac{h}{2 \cdot W_2} - 0.375 \cdot \frac{W_1}{h} \cdot \left(1 - \frac{W_2}{h}\right)\right)$$

$$Y = 165.6 \cdot \frac{W_2}{h} + 31.2\sqrt{\frac{W_2}{h}} - 11.8 \cdot \left(\frac{W_2}{h}\right)^2$$

$$L_1 = 10^{-9} \cdot h \cdot \left(Y \cdot \frac{W_1}{h} - 32 \cdot \frac{W_2}{h} + 3\right) \cdot \left(\frac{h}{W_1}\right)^{1.5}$$

$$L_3 = 10^{-9} \cdot h \cdot \left(5 \cdot \frac{W_2}{h} \cdot \cos\left(\frac{\pi}{2} \cdot \left(1.5 - \frac{W_1}{h}\right)\right) - \left(1 + \frac{7 \cdot h}{W_1}\right) \cdot \frac{h}{W_2} - 337.5\right)$$

The equation of L_2 is obtained from the one of L_1 by exchanging the indices (W_1 and W_2). Note that L_3 is negative, so the model is unphysical without external microstrip lines. The above-mentioned equations are accurate to within 5% for $0.3 \le W_1/h \le 3$ and

$0.1 \le W_2/h \le 3$ (value of C_1 ... C_4) or for $0.5 \le W_{1,\,2}/h \le 2$ (value of L_1.... L_3), respectively.

Some improvements to this model include:

- Comparisons with real life show that the value of L_3 is too large. Multiplying it by 0.8 leads to much better results.
- The model can be expanded for substrates with $\varepsilon_r \ne 9.9$ by modifying the values of the capacitances, as follows:

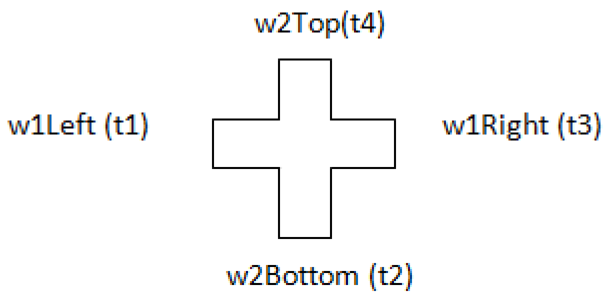
$$Cx = Cx(\varepsilon_r = 9.9) \cdot \frac{Z_0(\varepsilon_r = 9.9,\, W = Wx)}{Z_0(\varepsilon_r = \varepsilon_{r,\,sub},\, W = Wx)} \cdot \sqrt{\frac{\varepsilon_{eff}(\varepsilon_r = \varepsilon_{r,\,sub},\, W = Wx)}{\varepsilon_{eff}(\varepsilon_r = 9.9,\, W = Wx)}}$$

where:

The equations of Z_0 and ε_{eff} are the ones from the microstrip lines.

© 1994-2020
All Rights Reserved.

The model assumes that there are only two distinct values of the trace width, `w1Left=w1Right` and `w2Bottom=w2Top`. Trace widths are defined as follows:



As an approximation, the geometric mean of the trace width,

`W1=sqrt(w1Left * w1Right), W2=sqrt(w2Top * w2Bottom)` is used if this condition is not satisfied.

Command-line help

```
spectre -h cross
```

CDF Parameters

`mcros` has no CDF parameters

Range of Usage

Denote the larger trace width as W1, and the smaller trace width as W2. The model has accuracy of approximately 5% if the following inequalities are satisfied:

$$0.3 \le W_1/h \le 3$$

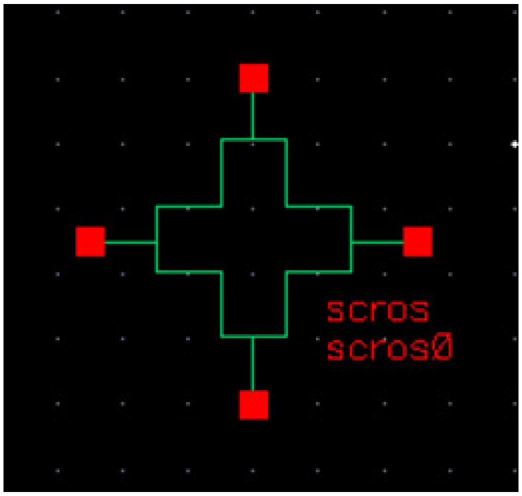
$$0.1 \le W_2/h \le 3$$

Example

```
mcros0 (t1 t2 t3 t4) cross
```


© 1994-2020 All Rights Reserved.

SYMBOL: SCROS



Note: The use of `scros` is not recommended in the February EAP release.

Stripline cross

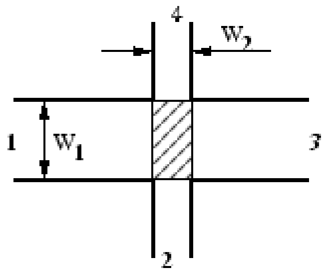
A cross instance must have exactly four transmission line instances connecting to its terminals, one transmission line per terminal. All four transmission lines should refer to the same stackup. It is an error to have instances of any other model connect to the terminals of a step.

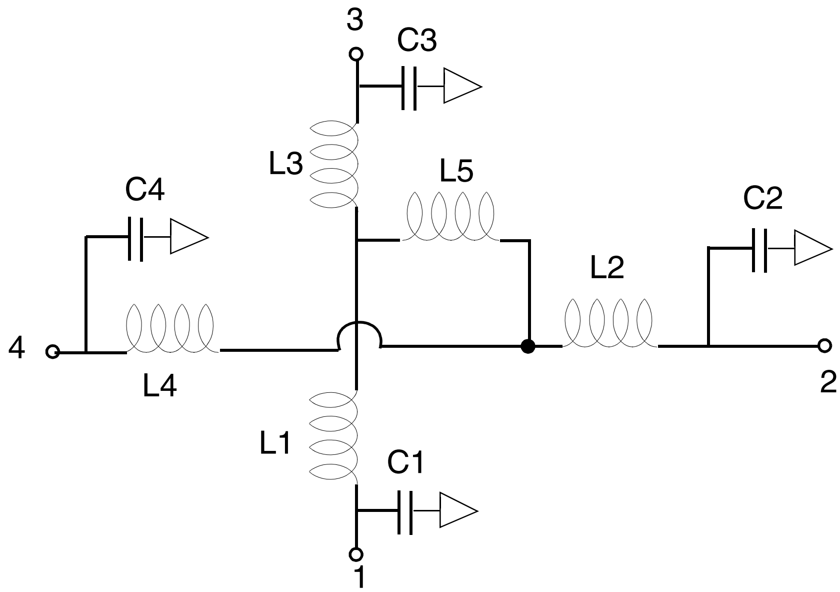
The geometric and material information is obtained automatically from the transmission line instances.

The frequency-domain model is an empirically based model which consists of a lumped equivalent circuit (five inductors and four shunt capacitors to ground). The equivalent circuit parameters are frequency-independent and are expressed as functions of trace width, dielectric thickness, and dielectric constant.

© 1994-2020 All Rights Reserved.

A diagram of the physical cross and the equivalent circuit is shown below.





The model of a microstrip cross instance shows the equivalent circuit (right-hand side) and the layout with dimensions (left-hand side). The hatched area in the layout marks the area modeled by the equivalent circuit. As can be seen, the model requires the microstrip width of line 1 and 3, as well as the one of line 2 and 4 to be equal to each other. In addition, the permittivity of the substrate must be $\varepsilon_r = 9.9$. The component values are calculated as follows:

$$X = \log 10\left(\frac{W_1}{h}\right) \cdot \left(86.6 \cdot \frac{W_2}{h} - 30.9 \cdot \sqrt{\frac{W_2}{h}} + 367\right) + \left(\frac{W_2}{h}\right)^3 + 74 \cdot \frac{W_2}{h} + 130$$

$$C_1 = C_2 = C_3 = C_4$$

$$= 10^{-12} W_1 \cdot \left(0.25 \cdot X \cdot \left(\frac{h}{W_1}\right)^{\frac{1}{3}} - 60 + \frac{h}{2 \cdot W_2} - 0.375 \cdot \frac{W_1}{h} \cdot \left(1 - \frac{W_2}{h}\right)\right)$$

$$Y = 165.6 \cdot \frac{W_2}{h} + 31.2\sqrt{\frac{W_2}{h}} - 11.8 \cdot \left(\frac{W_2}{h}\right)^2$$

$$L_1 = 10^{-9} \cdot h \cdot \left(Y \cdot \frac{W_1}{h} - 32 \cdot \frac{W_2}{h} + 3\right) \cdot \left(\frac{h}{W_1}\right)^{1.5}$$

$$L_3 = 10^{-9} \cdot h \cdot \left(5 \cdot \frac{W_2}{h} \cdot \cos\left(\frac{\pi}{2} \cdot \left(1.5 - \frac{W_1}{h}\right)\right) - \left(1 + \frac{7 \cdot h}{W_1}\right) \cdot \frac{h}{W_2} - 337.5\right)$$

© 1994-2020
All Rights Reserved.

The equation of L_2 is obtained from the one of L_1 by exchanging the indices (W_1 and W_2). Note that L_3 is negative, so the model is unphysical without external microstrip lines. The above-mentioned equations are accurate to within 5% for $0.3 \le W_1/h \le 3$ and $0.1 \le W_2/h \le 3$ (value of C_1 ... C_4) or for $0.5 \le W_{1,\,2}/h \le 2$ (value of L_1.... L_3), respectively.

Some improvements to this model include:

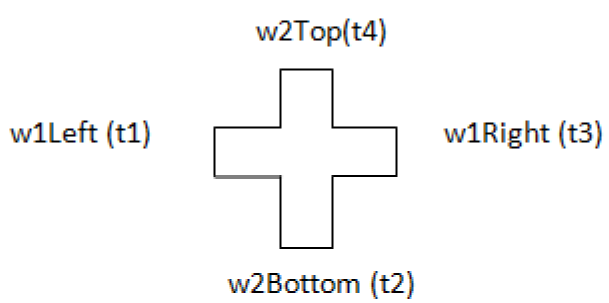
- Comparisons with real life show that the value of L_3 is too large. Multiplying it by 0.8 leads to much better results.
- The model can be expanded for substrates with $\varepsilon_r \ne 9.9$ by modifying the values of the capacitances, as follows:

$$Cx = Cx(\varepsilon_r = 9.9) \cdot \frac{Z_0(\varepsilon_r = 9.9,\, W = Wx)}{Z_0(\varepsilon_r = \varepsilon_{r,\,sub},\, W = Wx)} \cdot \sqrt{\frac{\varepsilon_{eff}(\varepsilon_r = \varepsilon_{r,\,sub},\, W = Wx)}{\varepsilon_{eff}(\varepsilon_r = 9.9,\, W = Wx)}}$$

  where:

  The equations of Z_0 and ε_{eff} are the ones from the microstrip lines.

The model assumes that there are only two distinct values of the trace width, `w1Left=w1Right` and `w2Bottom=w2Top`. Trace widths are defined as follows:



As an approximation, the geometric mean of the trace width,

`W1=sqrt(w1Left * w1Right), W2=sqrt(w2Top * w2Bottom)` is used if this condition is not satisfied.

Command-line help

```
spectre -h cross
```


© 1994-2020
All Rights Reserved.

CDF Parameters

`scros` has no CDF parameters

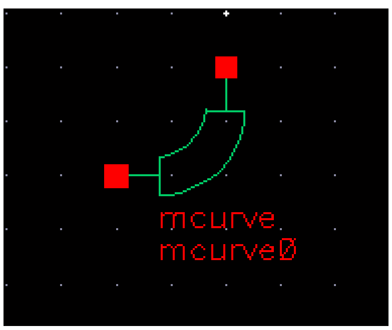
Range of Usage

The simulation frequency f should satisfy the equation `F[GHz] < Zo/B [mm]`, where `Zo` is the characteristic impedance, and B is the dielectric thickness (reference plane separation), in millimeters.

Example

```
scros0 (t1 t2 t3 t4) cross
```

SYMBOL: MCURVE



Microstrip curve

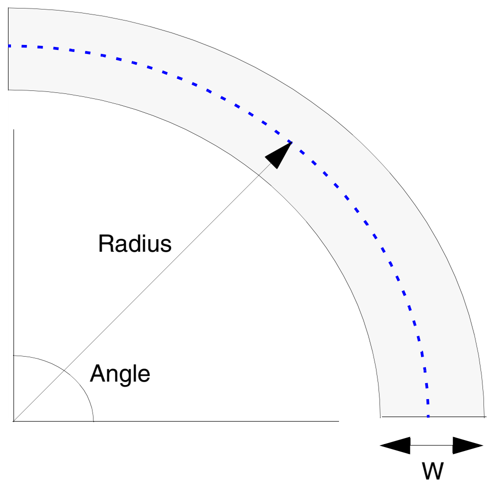
A curve instance must have exactly two transmission line instances connecting to its terminals, one per terminal. Both transmission lines should refer to the same stackup. It is an error to have instances of any other models connect to the terminals of a curve.

The geometric and material information is obtained automatically from the transmission line instances.

The frequency-domain model is an empirically based model which consists of an equivalent transmission line.

© 1994-2020
All Rights Reserved.

The geometry of the mcurve model is defined in the figure below.



The default value of the `radius` parameter in this model is 0. This signals the model to set the radius `r=2W`, where `W` is the trace width.

Command-line help

```
spectre -h curve
```

CDF Parameters

Label	Parameter	Type	Units	Default
Angle of curve(degrees)	angle	Double	Degrees	90
Radius	radius	Double	Meter	0

Range of usage

The ratio of the trace width W to the substrate thickness h should satisfy the inequality

- $0.01 \le W/h \le 100$
- $-180 \le Angle \le 100$

Due to the symmetry, the negative and positive values of Angle yield the same answer.

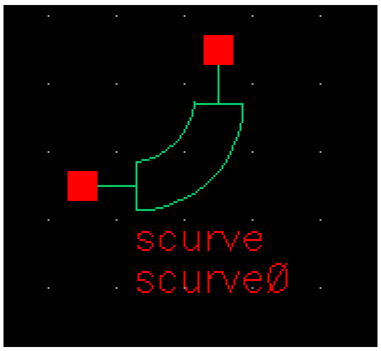
The radius r should be greater than `W/2` (r=0 actually means to set `r=2W` and is also allowed).

© 1994-2020 All Rights Reserved.

Example

```
mcurve0 (net1 net2) curve
```

SYMBOL: SCURVE



Stripline curve

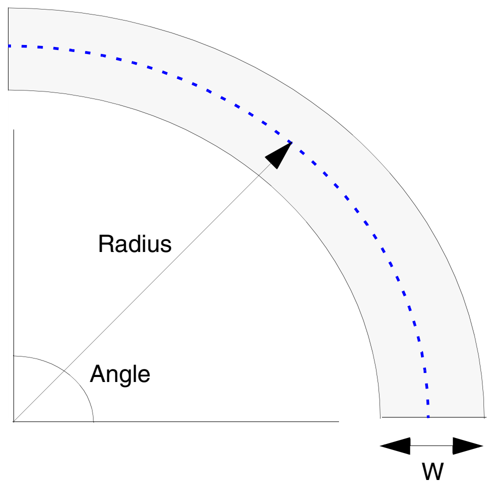
A curve instance must have exactly two transmission line instances connecting to its terminals, one per terminal. Both transmission lines should refer to the same stackup. It is an error to have instances of any other models connect to the terminals of a curve.

The geometric and material information is obtained automatically from the transmission line instances.

The frequency-domain model is an empirically based model which consists of an equivalent transmission line.

© 1994-2020 All Rights Reserved.

The geometry of the mcurve model is defined in the figure below.



The default value of the 'radius' parameter in this model is 0. This signals the model to set the radius `r=2W`, where `W` is the trace width.

Command-line help

```
spectre -h curve
```

CDF Parameters

Label	Parameter	Type	Units	Default
Angle of curve(degrees)	angle	Double	Degrees	90
Radius	radius	Double	Meter	0

Range of usage

The ratio of the trace width W to the substrate thickness h should satisfy the inequality

- $0.01 \leq W/h \leq 100$
- $-180 \leq Angle \leq 180$

Due to the symmetry, the negative and positive values of Angle yield the same answer.

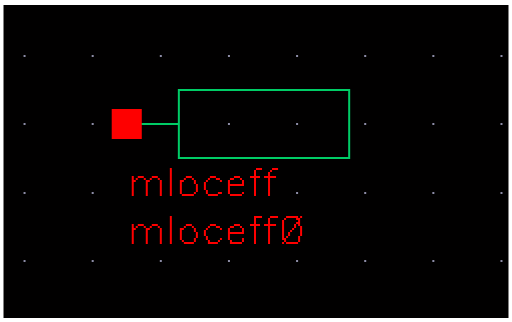
The radius r should be greater than `W/2` (r=0 actually means to set r=2W and is also allowed).

© 1994-2020 All Rights Reserved.

Example

```
scurve0 (net1 net2) curve
```

SYMBOL: MLOCEFF



Microstrip Open Circuit End Effect

The open end effect model mloceff accounts for the end effect of the microstrip transmission line (such as fringing capacitance). The typical usage involves connecting it to a microstrip transmission line. The `mloceff` model itself is not a transmission line model, and it has no length.

An `mloceff` instance must have exactly one transmission line instance connecting to its terminal. It is an error to have instances of any other models connect to the terminals of a curve.

The geometric and material information is obtained automatically from the transmission line instances.

The frequency-domain model is the transmission line length extension as described by Eqn. (4) in *E. Hammerstad, "Computer-Aided Design of Microstrip Couplers with Accurate Discontinuity Models", IEEE MTT-S Symp. Dig., p. 54, 1981*.

Command-line help

```
spectre -h loc
```

CDF Parameters

`mloceff` has no CDF parameters

© 1994-2020 All Rights Reserved.

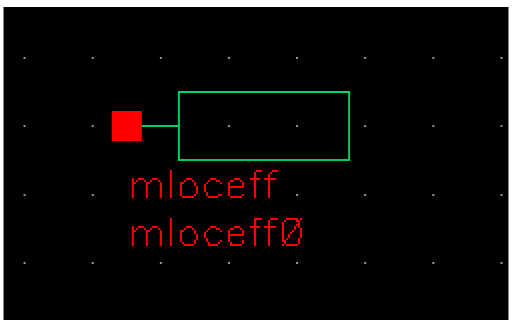
Range of usage

The model is sufficiently accurate (within 3%) for the ratio `W/h <3`, where `W` is the trace width, and `h` is the substrate thickness.

Example

```
mloceff0 (net1 net2) loc
```

SYMBOL: SLOCEFF



Stripline Open Circuit End Effect

The open end effect model `mloceff` accounts for the end effect of the microstrip transmission line (such as fringing capacitance). The typical usage involves connecting it to a microstrip transmission line. The `mloceff` model itself is not a transmission line model, and it has no length.

An `mloceff` instance must have exactly one transmission line instance connecting to its terminal. It is an error for instances of any other model to connect to the terminals of a curve.

The geometric and material information is obtained automatically from the transmission line instances.

The frequency-domain model is the transmission line length extension as described by Eqn. (6.2) in *K. C. Gupta, R. Garg, R. Chadha, Computer-Aided Design of Microwave Circuits, Artech House, 1981, p. 187*.

Command-line help

```
spectre -h loc
```


© 1994-2020 All Rights Reserved.

CDF Parameters

`sloceff` has no CDF parameters

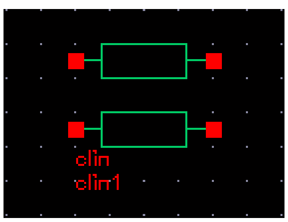
Range of usage

The ratio of trace with to the dielectric substrate thickness should not exceed 3.

Example

```
sloceff0 (net1 net2) loc
```

SYMBOL: CLIN



Lossless 2-Conductor Coupled Line

`clin` represents an ideal two-conductor coupled transmission line. `clin` is specified in terms of its even and odd-mode characteristic impedance, even and odd-mode electrical length, and the frequency at which the electrical lengths are specified.

© 1994-2020 All Rights Reserved.

Calculation procedure

Letting θ_e and θ_0 represent the even and odd-mode electrical lengths in degrees, f represent the frequency at which the electrical lengths are measured, and letting the physical length of the line be 1m, the even and odd mode phase velocities v_e and v_o are:

$$\nu_e = \frac{360f}{\theta_e}$$

$$\nu_o = \frac{360f}{\theta_o}$$

If the inductance and capacitance matrices of the line are expressed as:

$$L = \begin{bmatrix} L_d & L_m \\ L_m & L_d \end{bmatrix}$$

$$C = \begin{bmatrix} C_d & -C_m \\ -C_m & C_d \end{bmatrix}$$

where `Cm` is positive, the matrix elements are given by:

$$L_D = \frac{1}{2}\left(\frac{Z_e}{v_e} + \frac{Z_o}{v_o}\right)$$

$$L_m = \frac{1}{2}\left(\frac{Z_e}{v_e} - \frac{Z_o}{v_o}\right)$$

$$C_d = \frac{1}{2}\left(\frac{1}{Z_e v_e} + \frac{1}{Z_o v_o}\right)$$

$$C_m = \frac{1}{2}\left(\frac{1}{Z_o v_o} - \frac{1}{Z_e v_e}\right)$$

© 1994-2020 All Rights Reserved.

`clin` is netlisted as Spectre `mtline` with inductance and capacitance matrices, as specified in the above equations.

Parameter conditions

The clin parameters must satisfy the following conditions:

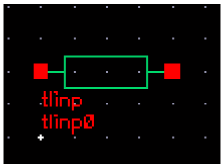
- $Z_e > Z_o > 0$
- $E_e \geq E_o > 0$
- $\frac{Z_e}{Z_o} > \frac{E_e}{E_o}$

CDF Parameters

Label	Type	Units	Default
Characteristic impedance of the even mode	Double	Ohms	75
Characteristic impedance of the off mode	Double	Ohms	25
Electrical length of the even mode	Double	Degrees	45
Electrical length of the odd mode	Double	Degrees	45
Reference frequency for the electrical length	Double	Hz	1G

© 1994-2020 All Rights Reserved.

SYMBOL: TLINP



Single-conductor behavioral transmission line

`tlinp` represents a lossy single-conductor transmission line. The line is defined in terms of the following CDF parameters:

CDF Parameters

Label	Type	Units	Default
Characteristic impedance	Double	Ohms	50
Physical length	Double	Meters	25.4
Effective dielectric constant	Double		2.1
Attenuation in dB/m	Double		0.001
Reference frequency for attenuation	Double	Hz	0
Loss tangent	Double		0.0002
Relative permeability	Double		1
Dielectric conductivity in Siemens	Double	Mhos	0
Dielectric loss freq range start	Double	Hz	
Dielectric loss freq range end	Double	Hz	

© 1994-2020 All Rights Reserved.

Notes

The characteristic impedance is specified at an infinite frequency (that is, ignoring loss).

Attenuation in dB/m specifies the conductor loss factor. Considering F to be the reference frequency for attenuation, there are two modes of loss behavior with respect to frequency - constant and square-root dependent. In particular, when F=0,

$$\alpha_c = A/20\log e$$

When `F>0`,

$$\alpha_c = A\sqrt{\frac{f}{F}}/20\log e$$

`tlinp` parameters are mapped to Spectre `mtline` parameters as follows:

```
tlinp1 (in out 0 0) mtline len=<Physical length> r=[(R)] l=[(L)] g=[(G0)] c=[(C)]
rskin=[(Rs)] gdloss=[(Gd)]
```

where

$$C = \frac{\sqrt{\mu_r}k}{cZ}$$

© 1994-2020 All Rights Reserved.

`L=CZ`^{2}

$$G_0 = \frac{C\sigma}{k\varepsilon_0}$$

$$G_d = 2\pi C \tan d$$

$$R = \begin{cases} \dfrac{2ZA}{20\log e}, F = 0 \\ 1e^{-9}, F > 0 \end{cases}$$

$$R_s = \begin{cases} 0, F = 0 \\ \dfrac{2ZA}{\sqrt{F}20\log e}, F > 0 \end{cases}$$

Where:

- μ_r is the relative permeability
- k is the specified effective dielectric constant
- c is the speed of light in vacuum
- σ is the specified dielectric conductivity in S
- ε_0 is the vacuum permittivity
- `tand` is the specified loss tangent
- A is the specified conductor attenuation in dB/m
- F is the specified reference frequency

Parameter Conditions

The `tlinp` parameters must satisfy the following conditions:

- $Z > 0$
- $K \geq 1$

© 1994-2020 All Rights Reserved.

- $A \geq 0$
- $F \geq 0$

© 1994-2020 All Rights Reserved.

Obsolete Components

Note: The components explained below are obsolete, and should not be used for new designs. They are preserved for existing designs.

The following are the obsolete elements contained in the RF transmission line library, `rfTlineLib`:

- Microstrip Components
  - microbend2 on page 381
  - microbend90 on page 383
  - microopenend on page 385
  - microstep on page 387
  - microstrip on page 389
- Stripline Components
  - stripbend90 on page 392
  - stripline on page 394

These components can be placed in a schematic to approximate the effect that the layout has on the operation of the circuit. When the schematic is netlisted, the tool generates an equivalent RLCG subcircuit for each `rfTlineLib` component and instantiates the new subcircuits in the design. This additional detail improves the accuracy of the simulation.

Placing the `rfTlineLib` components in the schematic requires you to provide information such as the widths and thicknesses of strips and the heights of substrates. The process of adding one of these components to the schematic opens a form where you provide the required information as CDF values.

© 1994-2020 All Rights Reserved.

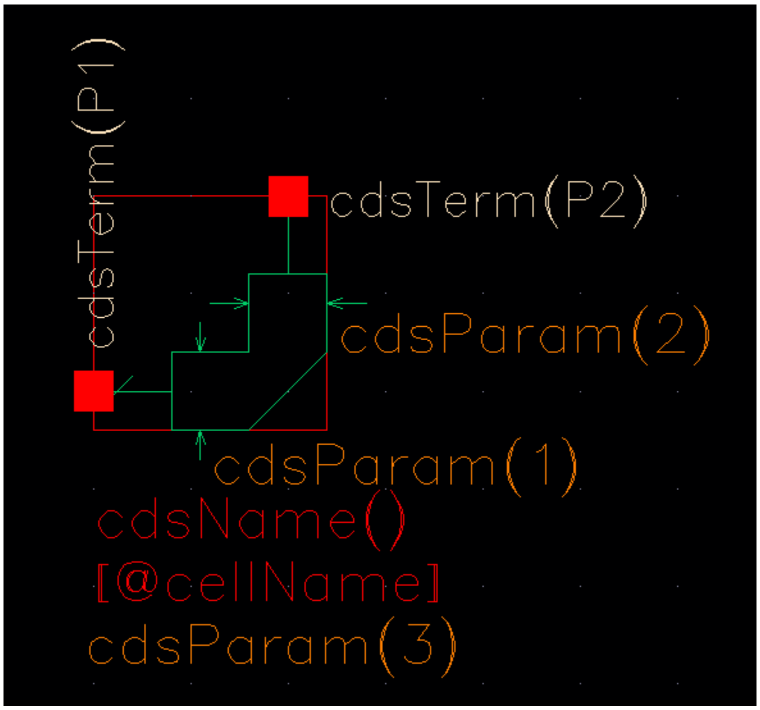
Microstrip Components

A microstrip consists of a conducting strip that is separated from a ground plate by a substrate. The dielectric on the other side of the conducting strip is typically air.

© 1994-2020 All Rights Reserved.

microbend2

This is a microstrip chamfered right-angle bend.

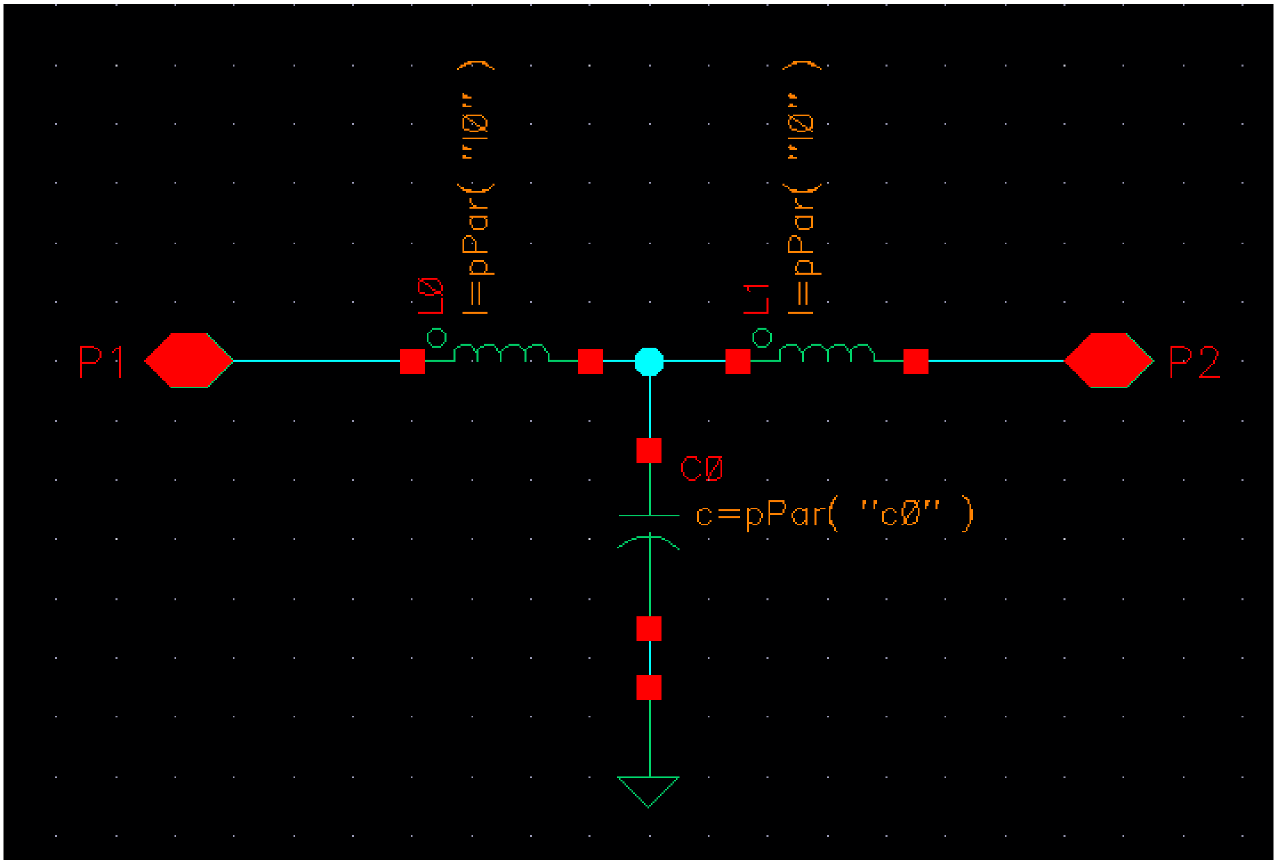


The parameters are:

`Dielectric constant of subs`	Relative dielectric constant of the substrate.
`Height of substrate`	Height of the substrate, in meters.
`Width of strip`	Width of the microstrip, in meters. The width is the same on both sides of the bend and, in the symbol, is labeled as both `cdsParam(1)` and `cdsParam(2)`.

© 1994-2020 All Rights Reserved.

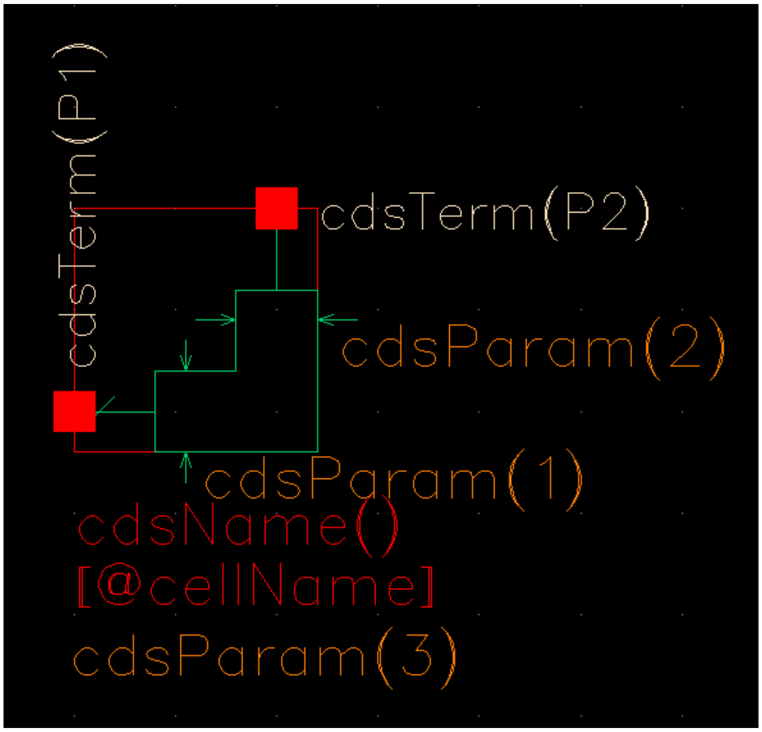
The equivalent circuit is this.



© 1994-2020
All Rights Reserved.

microbend90

This is a microstrip right-angle bend.

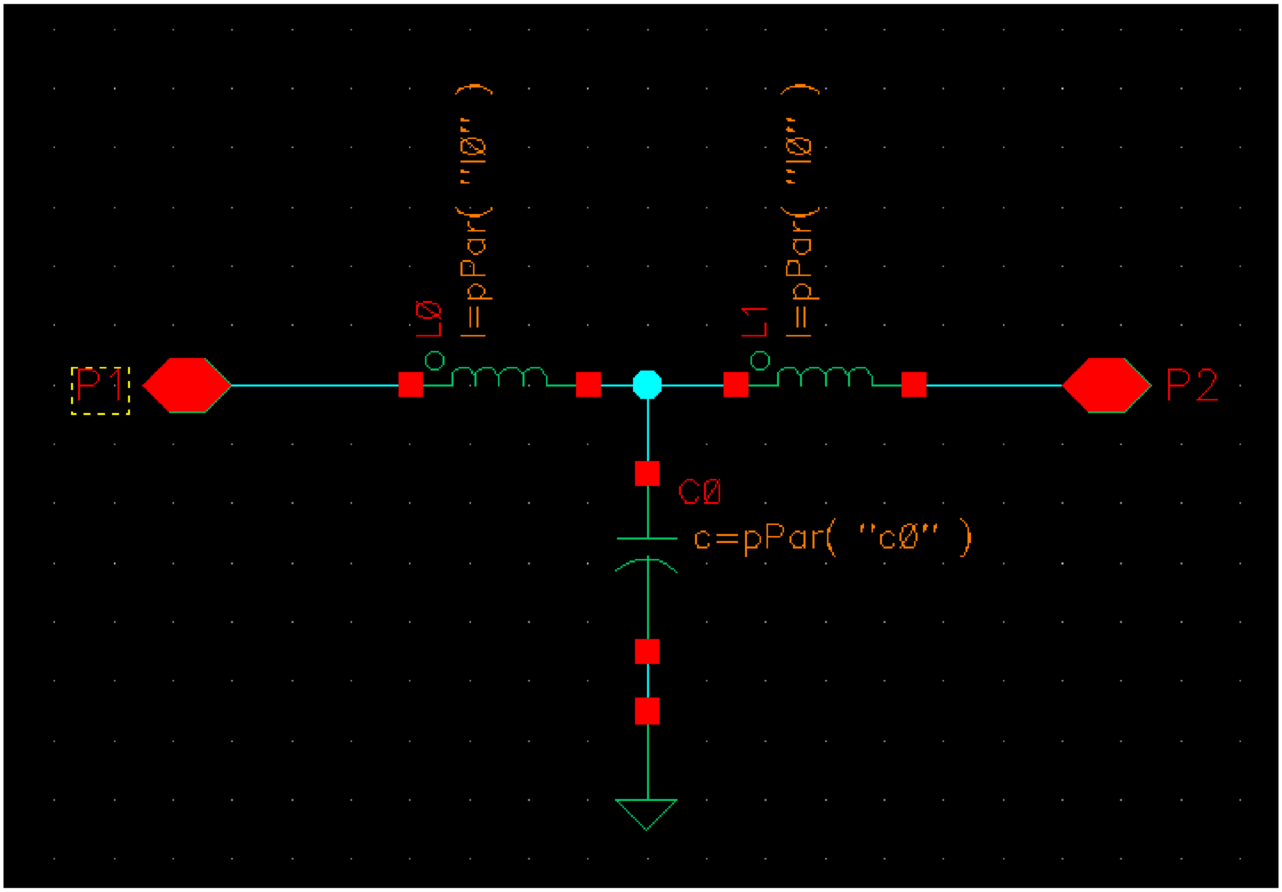


The parameters are:

`Dielectric constant of subs`	Relative dielectric constant of the substrate.
`Height of substrate`	Height of the substrate, in meters.
`Width of strip`	Width of the microstrip, in meters. The width is the same on both sides of the bend and, in the symbol, is labeled as both `cdsParam(1)` and `cdsParam(2)`.

© 1994-2020 All Rights Reserved.

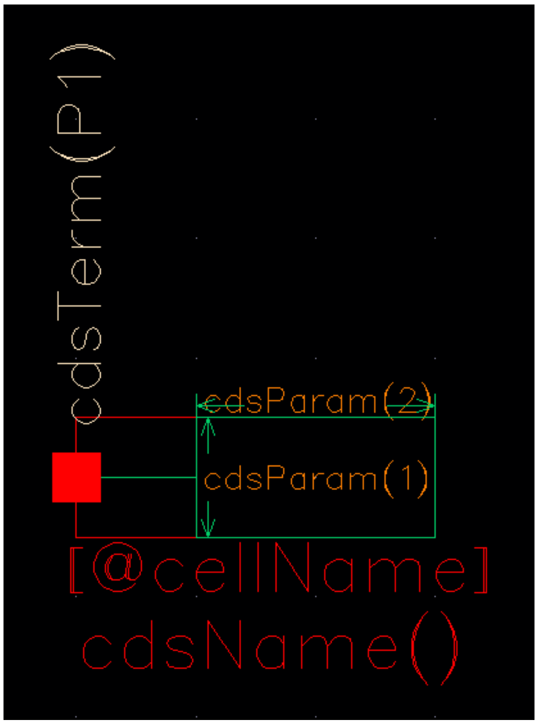
The equivalent circuit is this.



© 1994-2020
All Rights Reserved.

microopenend

This is a microstrip open end line.

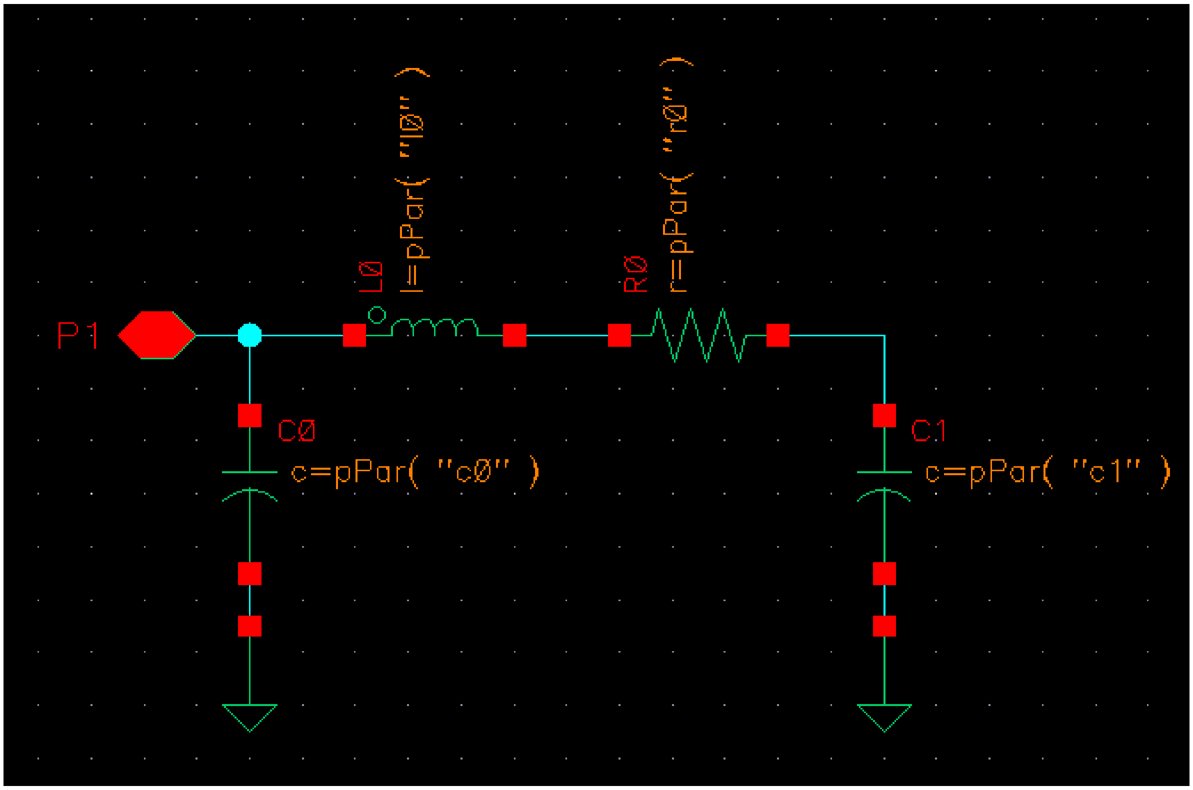


The parameters are:

`Electrical conductivity`	Electrical conductivity of the microstrip.
`Thickness of strip`	Thickness of the microstrip, in meters.
`Dielectric content of subs`	Relative dielectric constant of the substrate.
`Height of substrate`	Height of the substrate.
`Width of strip`	Width of the microstrip, in meters. In the symbol, this is labeled `cdsParam(1)`.
`Length of strip`	Length of the microstrip, in meters. In the symbol, this is labeled `cdsParam(2)`.
`Operating frequency`	Operating frequency.

© 1994-2020 All Rights Reserved.

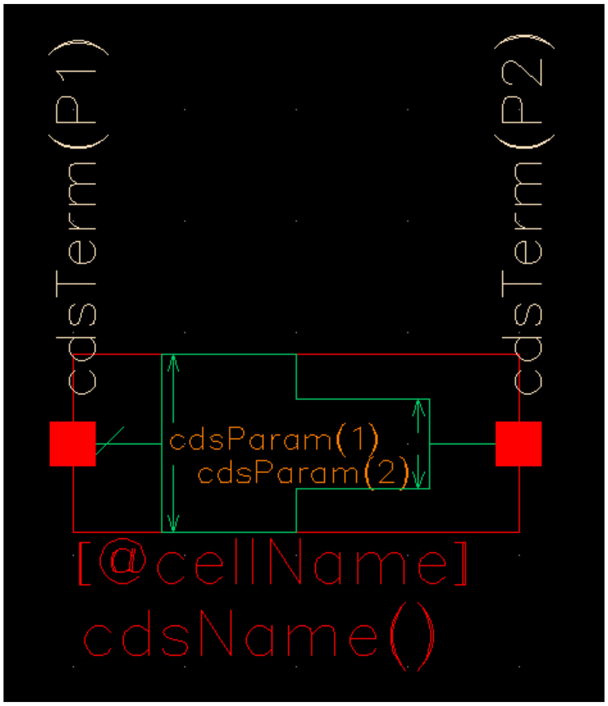
The equivalent circuit is this.



© 1994-2020
All Rights Reserved.

microstep

This component steps up or down the width of a microstrip.

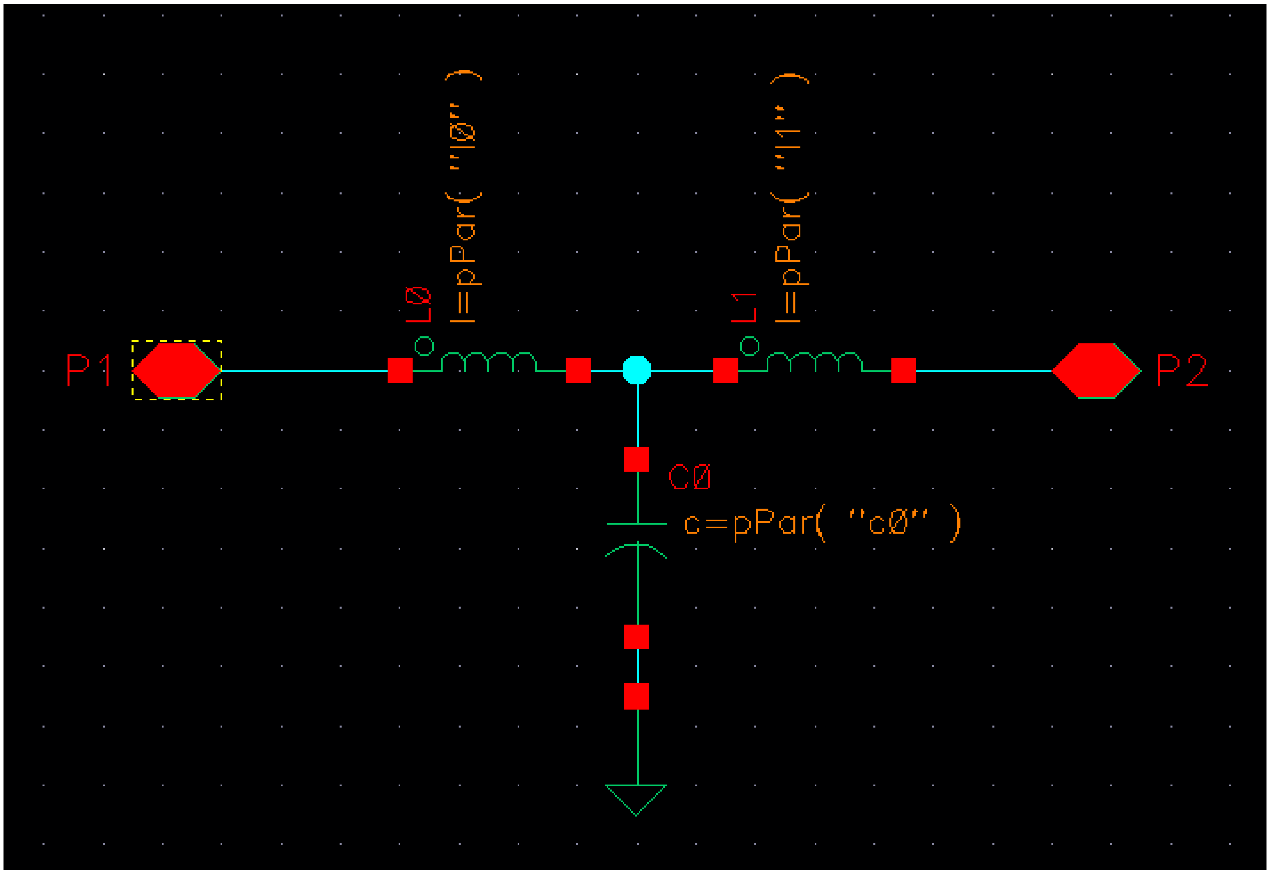


The parameters are:

`Dielectric constant of subs`	Relative dielectric constant of the substrate.
`Width of strip (W1)`	Width of one side of the microstrip, in meters. In the symbol, this is labeled `cdsParam(1)`.
`Width of strip (W2)`	Width of the other side of the microstrip, in meters. In the symbol, this is labeled `cdsParam(2)`.
`Height of substrate`	Height of the substrate.
`Operating frequency`	Operating frequency.
`Thickness of strip`	Thickness of the microstrip.

© 1994-2020 All Rights Reserved.

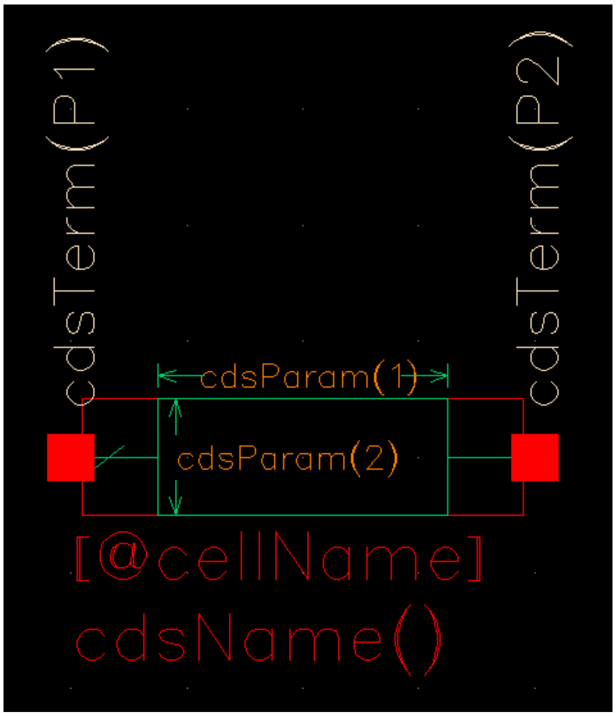
The equivalent circuit is this.



© 1994-2020
All Rights Reserved.

microstrip

This is the basic microstrip.

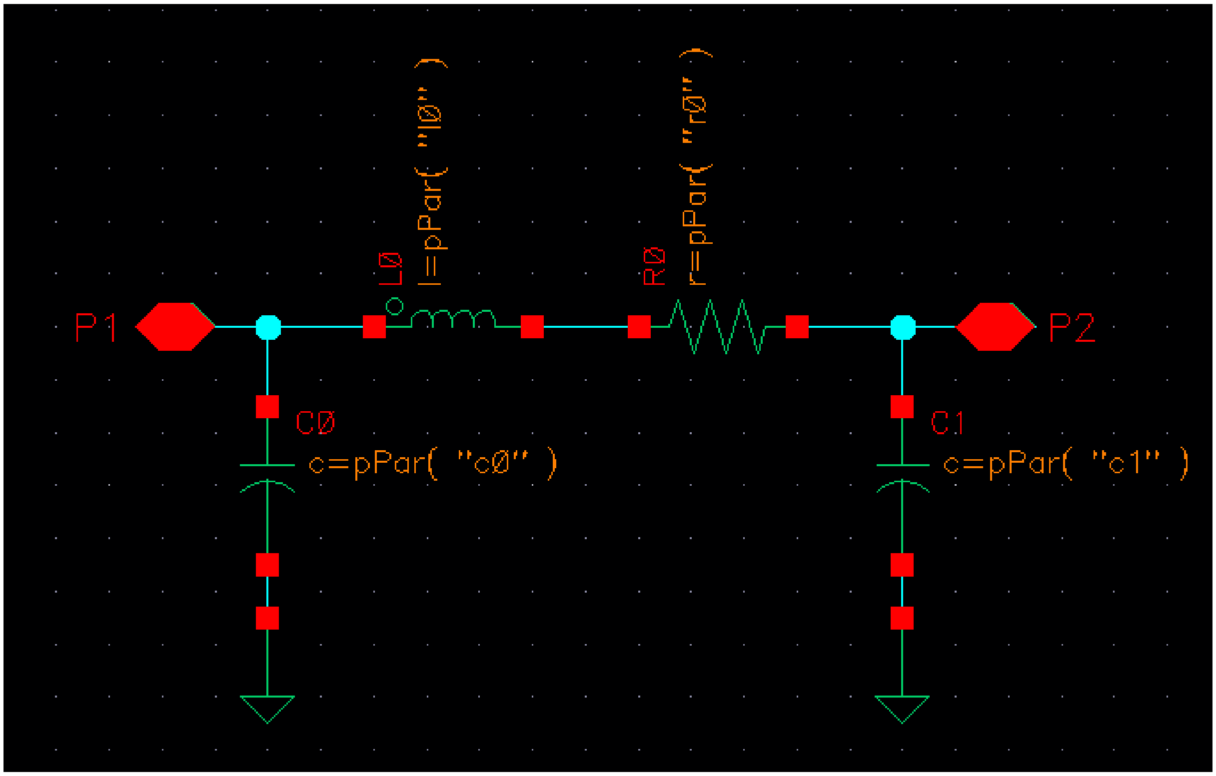


The parameters are:

`Electrical conductivity`	Electrical conductivity of the microstrip.
`Thickness of strip`	Thickness of the microstrip, in meters.
`Dielectric constant of subs`	Relative dielectric constant of substrate.
`Height of substrate`	Height of the substrate.
`Width of strip`	Width of the microstrip, in meters. In the symbol, this is labeled `cdsParam(2)`.
`Length of strip`	Length of the microstrip, in meters. In the symbol, this is labeled `cdsParam(1)`.
`Operating frequency`	Operating frequency.

© 1994-2020 All Rights Reserved.

The equivalent circuit is this.



© 1994-2020
All Rights Reserved.

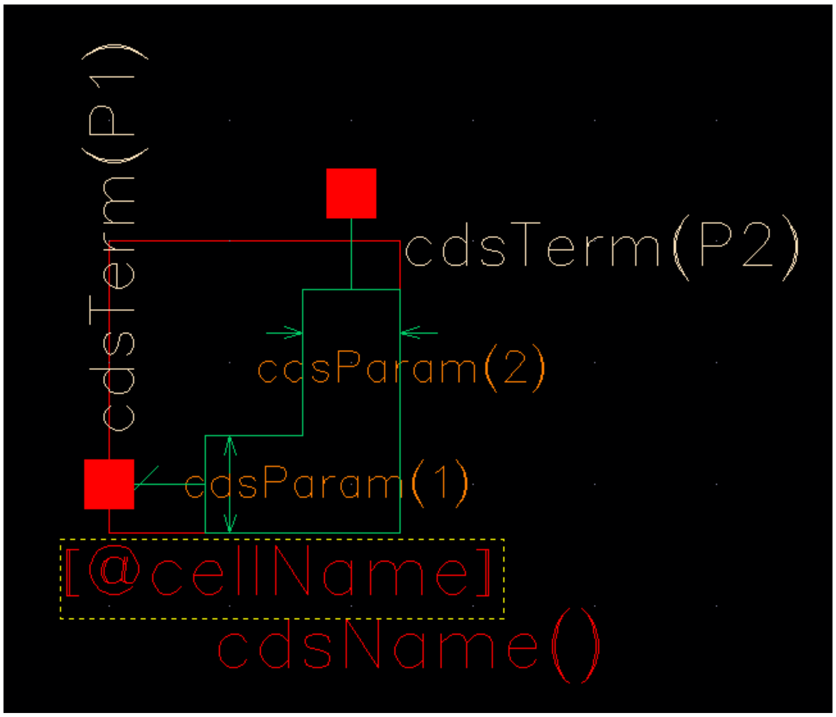
Stripline Components

A stripline consists of a conducting strip that is sandwiched between two ground planes but separated from each of them by a substrate layer.

© 1994-2020 All Rights Reserved.

stripbend90

This is a stripline right-angle bend.

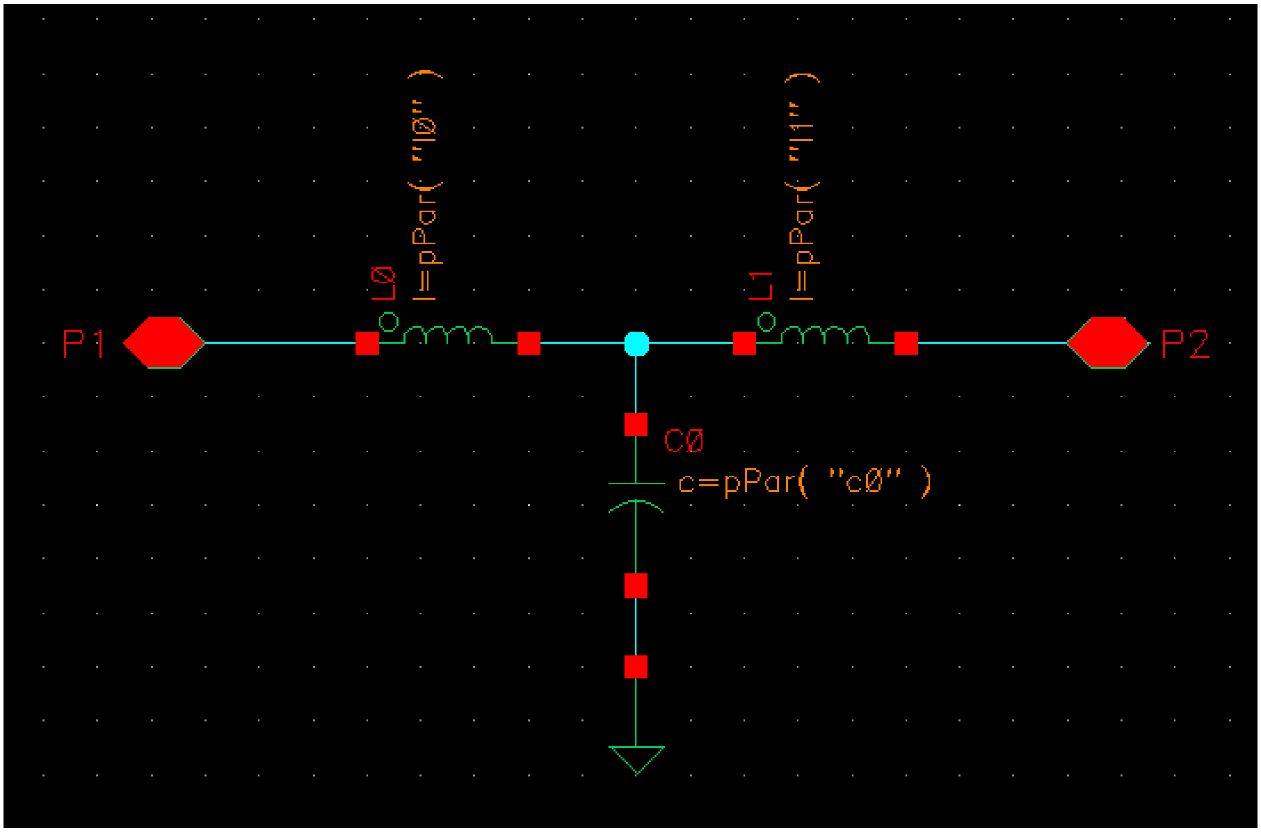


The parameters are:

`Dielectric constant of subs`	Relative dielectric constant of the substrate.
`Height of substrate`	Height of the substrate, in meters.
`Thickness of strip`	Thickness of the stripline, in meters.
`Width of strip`	Width of the stripline, in meters. The width is the same on both sides of the bend and, in the symbol, is labeled as both `cdsParam(1)` and `cdsParam(2)`.
`Operating frequency`	Operating frequency.

© 1994-2020
All Rights Reserved.

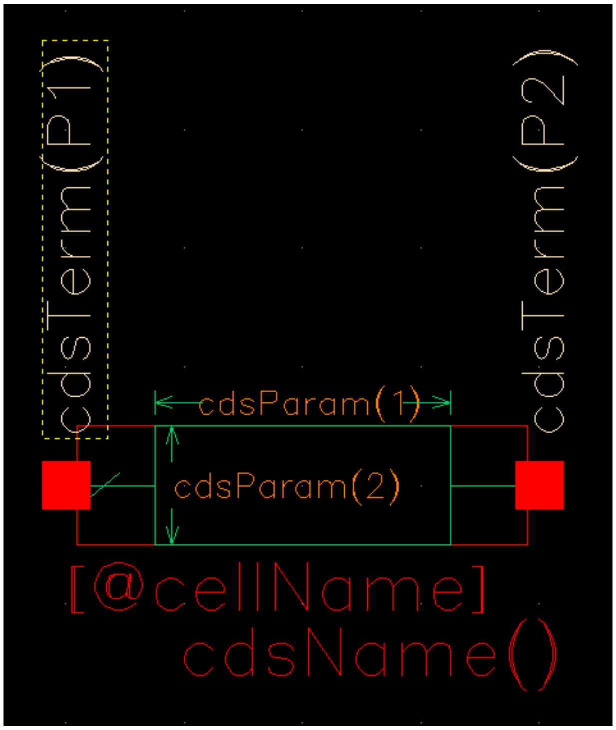
The equivalent circuit is this.



© 1994-2020 All Rights Reserved.

stripline

This is the basic stripline.

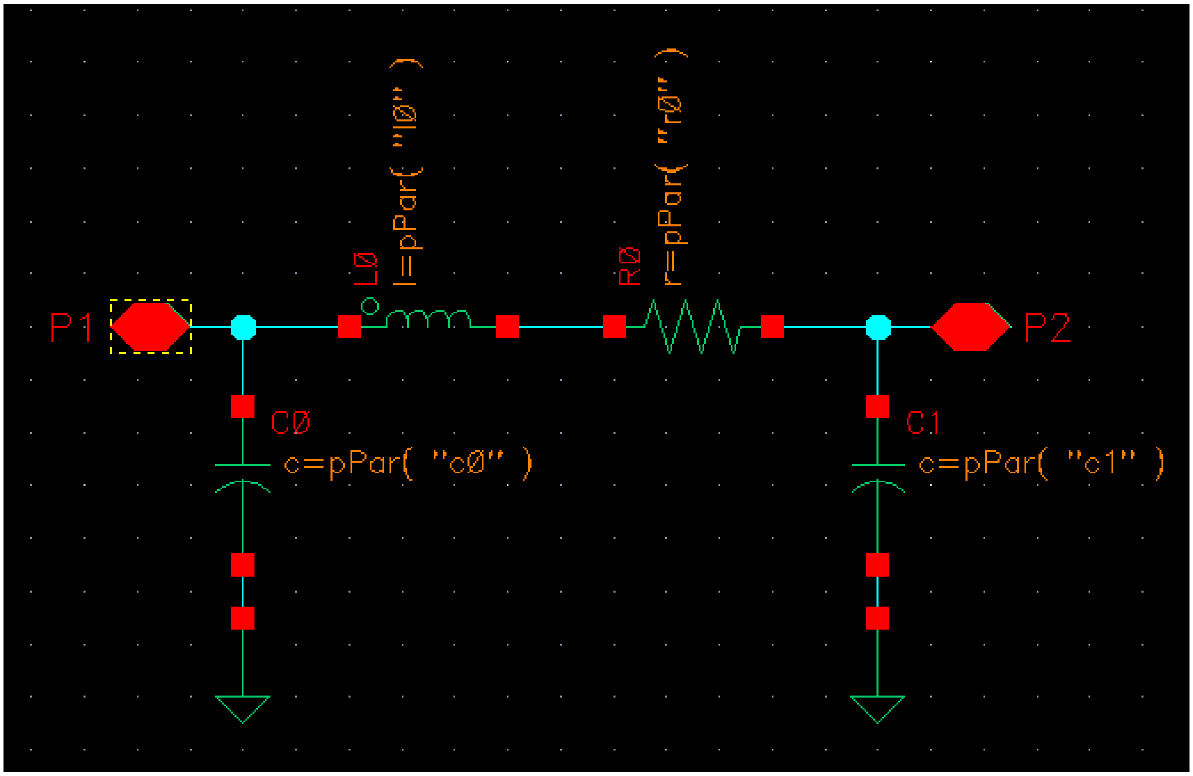


The parameters are:

`Electrical conductivity`	Electrical conductivity of the stripline.
`Thickness of strip`	Thickness of the stripline, in meters.
`Dielectric constant of subs`	Relative dielectric constant of the substrate.
`Height of substrate`	Height of the substrate, in meters.
`Width of strip`	Width of the stripline, in meters. In the symbol, this is labeled `cdsParam(2)`.
`Length of strip`	Length of the stripline, in meters. In the symbol, this is labeled `cdsParam(1)`.
`Operating frequency`	Operating frequency.

© 1994-2020 All Rights Reserved.

The equivalent circuit is this.



© 1994-2020
All Rights Reserved.

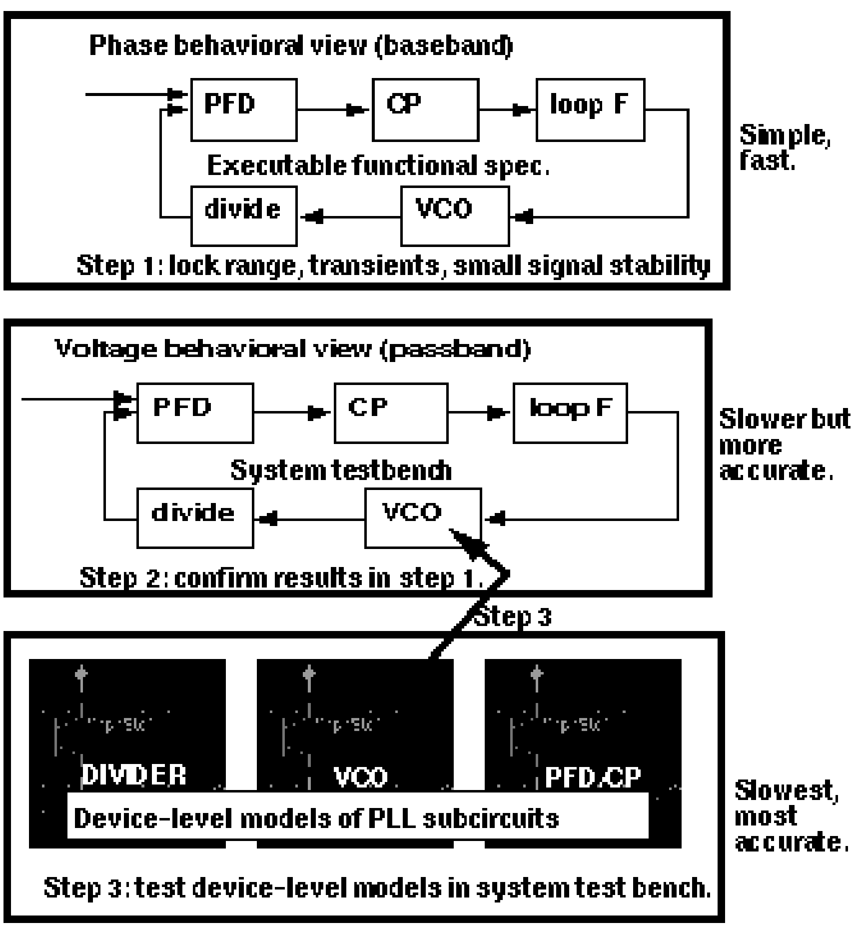
© 1994-2020
All Rights Reserved.

3

Introduction to the PLL library

The models in the phase lock loop (PLL) library support top-down design of PLLs. Figure 3-1 shows the three steps of the design flow. This appendix describes the first step in detail; all three steps are described briefly.

Figure 3-1 PLL Top-Down Design Flow



1. The first step in Figure 3-1 on page 397 is to develop an executable specification. The executable specification is an arrangement of fast behavioral models that permits fast architectural studies to separate specification and implementation issues. The executable specification contains baseband models [1,2,3,4,5,6,7,8,9]. (Reference [1] uses the terms “baseband” and “bandpass” explicitly.)

   These baseband models suppress clocks and RF/IF carriers. Some literature refers to PLL baseband models as “relative phase” or “phase-domain” models [2]. This appendix uses the latter term. Phase-domain PLL models are exceptionally fast, capture the important non-linear mechanisms, and can be linearized directly for AC analysis.

2. The second step in Figure 3-1 on page 397 is to translate the executable specification into a system testbench. The system testbench, unlike the executable specification, is composed of passband models [1]. This Appendix refers to passband models as voltage-domain models because they simulate voltages you can observe in a laboratory.

3. Comparing voltage- and phase-domain voltage-controlled oscillator (VCO) models highlights the difference between the two models. The output of a voltage-domain VCO model is a clock voltage, a periodic signal. The output of a phase-domain VCO model is a voltage numerically equal to phase. If you unwrap the VCO phase, in steady state, it ramps up indefinitely. Unwrapped phase is not periodic. Voltage-domain models describe non-linear effects related to the shapes of the actual RF waveforms. Such waveform effects include spurs and harmonic locking. Harmonic locking occurs when the PLL locks on to a harmonic of the reference.

4. Phase-domain models do not simulate waveform effects. The system testbench is more accurate than the executable specification, but it is still behavioral. Equipped only with behavioral voltage-domain models, the testbench does not simulate device-level effects associated with specific implementations. Examples of such implementation effects are interstage loading, improper bias, and device parasitics.

5. The last step in Figure 3-1 on page 397 is to gradually replace the behavioral models in the system testbench with device-level models, one or two blocks at a time. Device-level models check for the previously mentioned implementation problems. The entire PLL is simulated at the device-level only as a final verification step because such simulations are very lengthy.

Models in the PLL library

The PLL library includes the following phase-domain models:

- Analog multiplier phase detector
- XOR phase detector with bipolar output

© 1994-2020 All Rights Reserved.

The XOR phase detector is not explicitly discussed here because it is very similar to the analog multiplier phase detector. The only difference is that the duty cycle-phase error transfer curve is triangular instead of sinusoidal.

- Three-state digital phase frequency detector (PFD)
- Charge pump (current source version)
- VCO tuning curve (analytic and tabular versions)
- Frequency divider
- Lock indicator

Introduction to the PLL Library Documentation

The primary system-level specifications captured by this first set of phase-domain models are acquisition time, lock and capture ranges [12], and phase margin. The PFD model also simulates backlash[8]. Backlash is sometimes called “deadband” effect. It is a limit cycle caused by the phase-frequency detector’s inability to linearly reduce its output pulse width to zero as phase error goes to zero.

The remainder of this appendix is divided into two main sections.

The first section introduces phase-domain modeling, describes a feature included to prevent DC convergence problems, and then shows you some examples of using phase-domain models.

The second section explains how to assemble a more complex PLL and discusses an example. The examples are introductory and are not a comprehensive discussion of all applications of phase-domain models.

Phase-Domain Model of a Simple PLL

Description

This PLL example, which is built around the simplest phase detector in the library, introduces the fundamentals of phase-domain modeling. Figure 3-2 on page 400 shows a voltage-domain model of the example and also some selected waveforms. The phase detector in this case is an ideal analog multiplier.

© 1994-2020 All Rights Reserved.

Figure 3-2 Voltage Domain Model

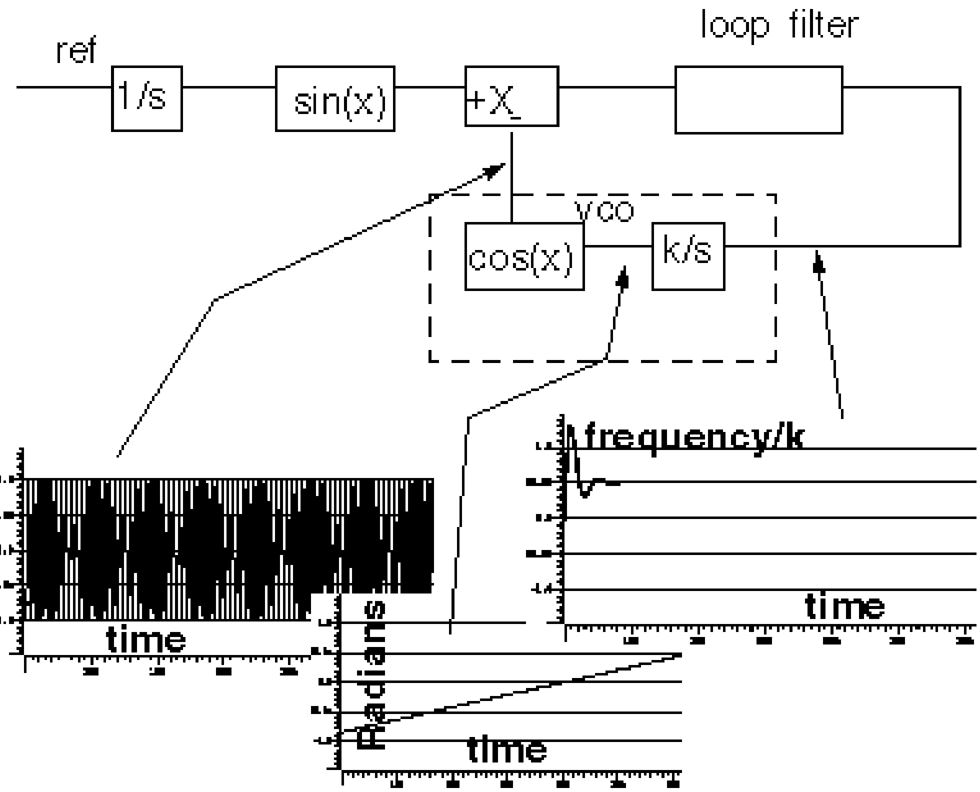


Figure 3-3 on page 401 shows the equivalent phase-domain model. The phase-domain model is based on the following trigonometric identity:

$\sin(\theta_1)*\cos(\theta_2) = (1/2)*[\sin(\theta_1+\theta_2)+\sin(\theta_1-\theta_2)]$

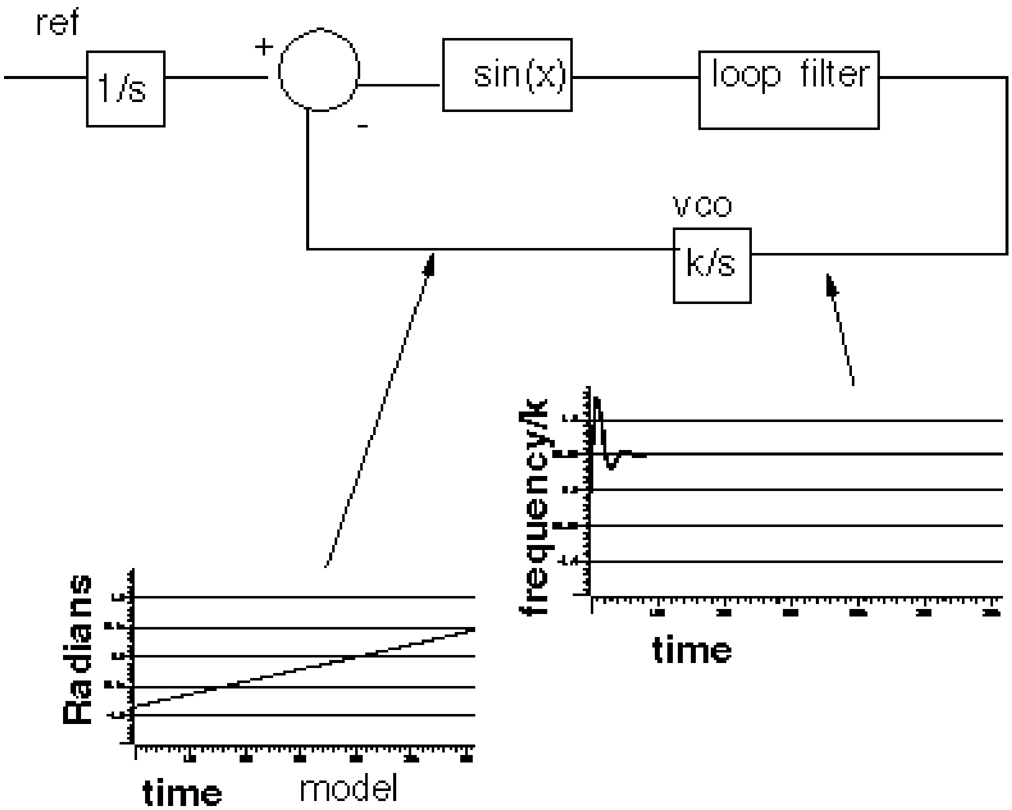
which, after filtering is approximately

$(1/2)*\sin(\theta_1-\theta_2)$

Note: $q_1+q_2=(w_1+w_2)*t$ and w_1+w_2 usually lie far beyond the filter's corner frequency.

© 1994-2020 All Rights Reserved.

Figure 3-3 Phase Domain Model

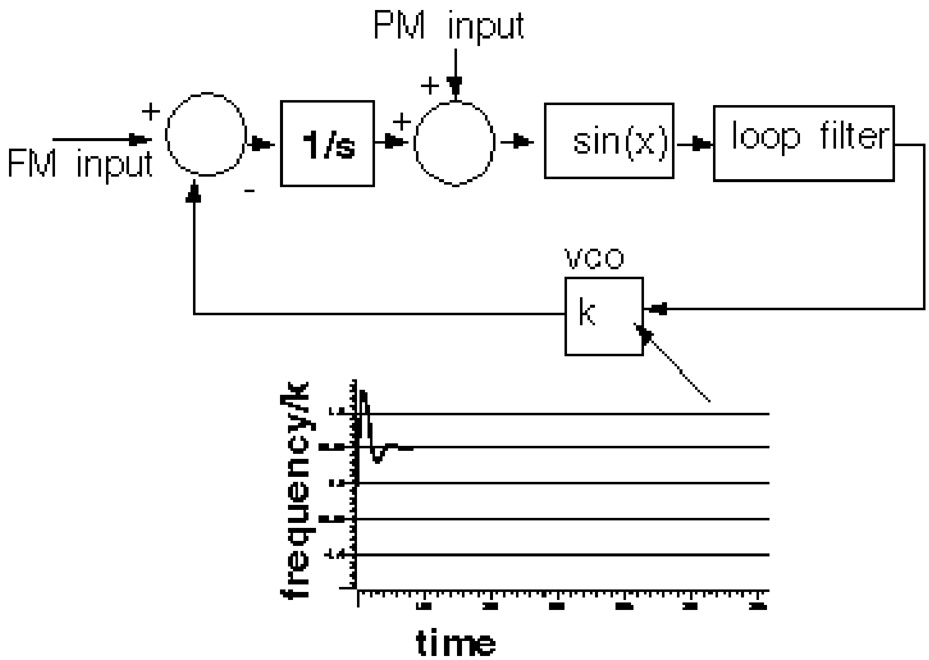


The phase and frequency waveforms from the phase-domain model match their voltage-domain counterparts, but the simulation runs faster because the oscillatory waveform is not explicitly simulated.

Combining the integrators, as shown in Figure 3-4 on page 402, eliminates the integrator outside the feedback loop which might cause a problem if you forget to specify an initial condition. Combining the integrators is also necessary if you build phase-domain models of phase-frequency detectors because, in this case, the non-linearity has memory (hysteresis).

© 1994-2020 All Rights Reserved.

Figure 3-4 A More Practical Phase Domain Model

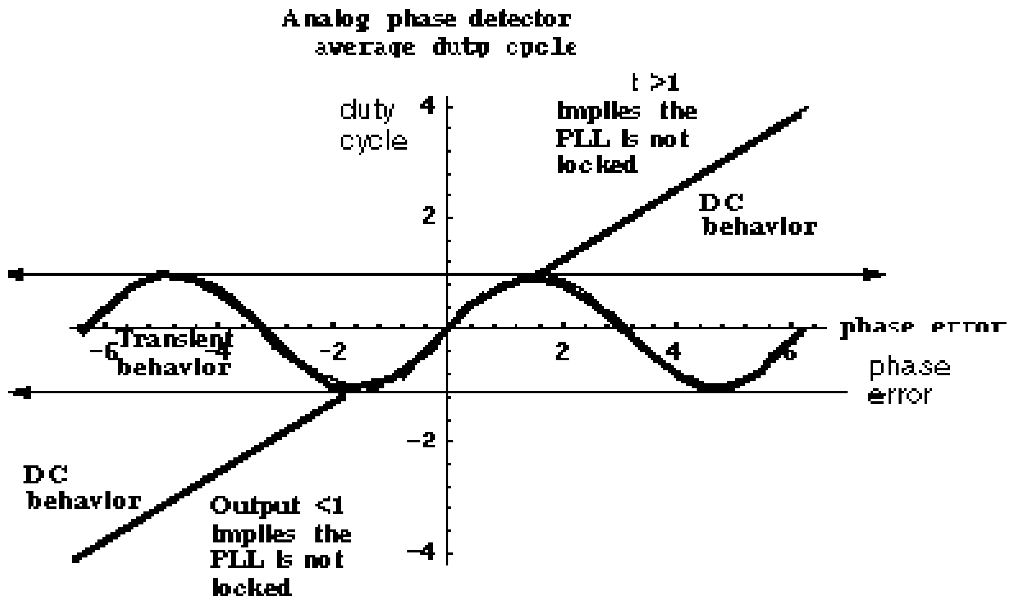


The models in both Figure 3-3 and Figure 3-4 can fail to achieve DC convergence because the phase detector model either has no DC operating point or because it has an infinite number of operating points.

The sinusoidal function in Figure 3-5 on page 403 is the phase detector transfer curve. The phase detector is the only non-linear element in this PLL model. For reasons associated with phase-frequency detectors, the phase detector output is called the *duty cycle*.

© 1994-2020 All Rights Reserved.

Figure 3-5 Phase Detector Transfer Curves



If the required duty cycle lies outside [-1,1], the loop is not locked in steady state. If the required duty cycle lies within [-1,1], there are an infinite number of possible phase errors. In either case, a Spectre® circuit simulator RF analysis (Spectre RF) simulation might not converge. The ability of Verilog-A© to perform different tasks for different analyses provides an elegant solution to the DC convergence problems and a quick way to map out lock range. Lock range is the range of input frequencies for which the PLL can maintain lock. (Some literature refers to lock range as hold-in range [8].)

The phase detector model uses the monotonic transfer curve for DC analysis and the true periodic transfer curve for transient analysis. The two transfer curves coincide when the phase error lies in the interval [$-\pi/2$, $\pi/2$]. If the required duty cycle lies within [-1,1], the monotonic transfer curve forces the steady-state phase error to the interval [$-\pi/2$, $\pi/2$], where the two curves coincide. The equilibrium point is *open-loop-stable*, meaning that at DC the loop gain is a positive real number. This is true because the slope of the transfer curve is positive over [$-\pi/2$, $\pi/2$]. The Nyquist stability criterion is therefore easier to apply. The DC analysis is general enough because only the phase error modulo 2π is of interest, and you usually care only about the open-loop-stable operating points. When the loop is not locked, the DC analysis computes a duty cycle with a magnitude greater than one. A duty cycle greater than one is clearly incorrect, but it is much easier to interpret than a convergence error. DC duty cycle is a lock indicator which can be used in a parametric DC analysis to sweep out lock range.

© 1994-2020
All Rights Reserved.

Example 1: Dynamic Test for Capture Range and Lock Range

The circuit used to dynamically test for capture range and lock is range is *example_analog_PD* in the *pllLib* library. *Capture range* is the range of input frequencies that the PLL can acquire from an unlocked state. Because acquisition of frequencies near the edge of the capture range involves a pull-in mechanism [1-12], measuring the capture range requires a transient analysis. You can measure capture and lock ranges by slowly sweeping the input frequency and observing the frequency at which the duty cycle begins and ends a long ramp [12]. You must skip the DC analysis to observe the capture limits. Figure 3-6 on page 404 plots the VCO control voltage against the input frequency voltage. The input frequency first ramps up and then down. A buffered auxiliary circuit responds to the duty cycle and adds 2.5 volts when the input frequency changes direction. This technique makes the plot easier to read because the forward and reverse sweeps occupy different parts of the vertical scale. In this example, lock range is from 1.36Khz to 3.4Khz, and the capture range is from 1.8Khz to 3Khz.

Figure 3-6 Lock Range and Capture Range

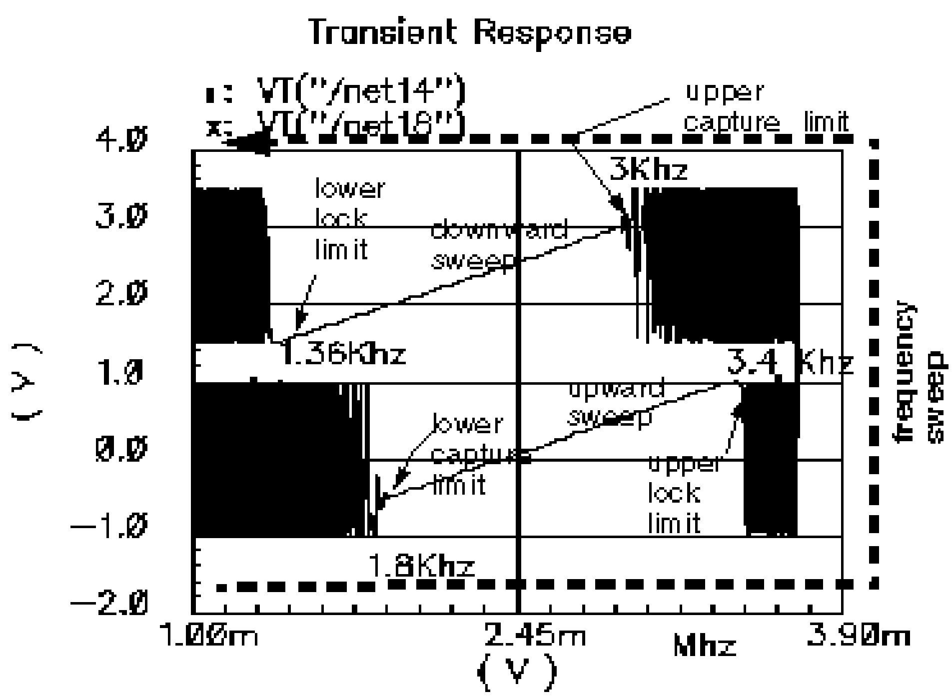
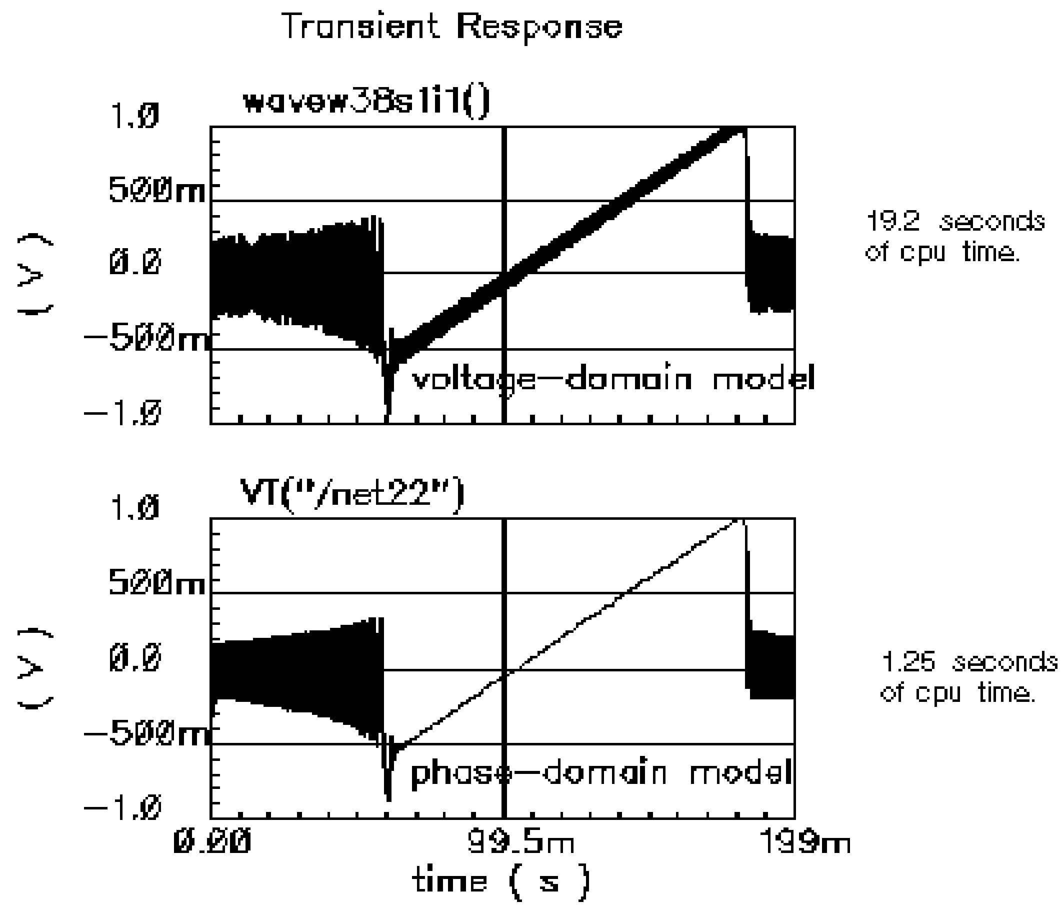


Figure 3-7 on page 405 compares VCO control voltages in the forward sweep when computed with voltage- or phase-domain models. The models produce similar results. In this example (2.5Khz center frequency), the phase-domain model is only about 20 times faster than the voltage-domain model.

All Rights Reserved.

Figure 3-7 VCO Control Voltages Computed with Voltage- and Phase-Domain Models



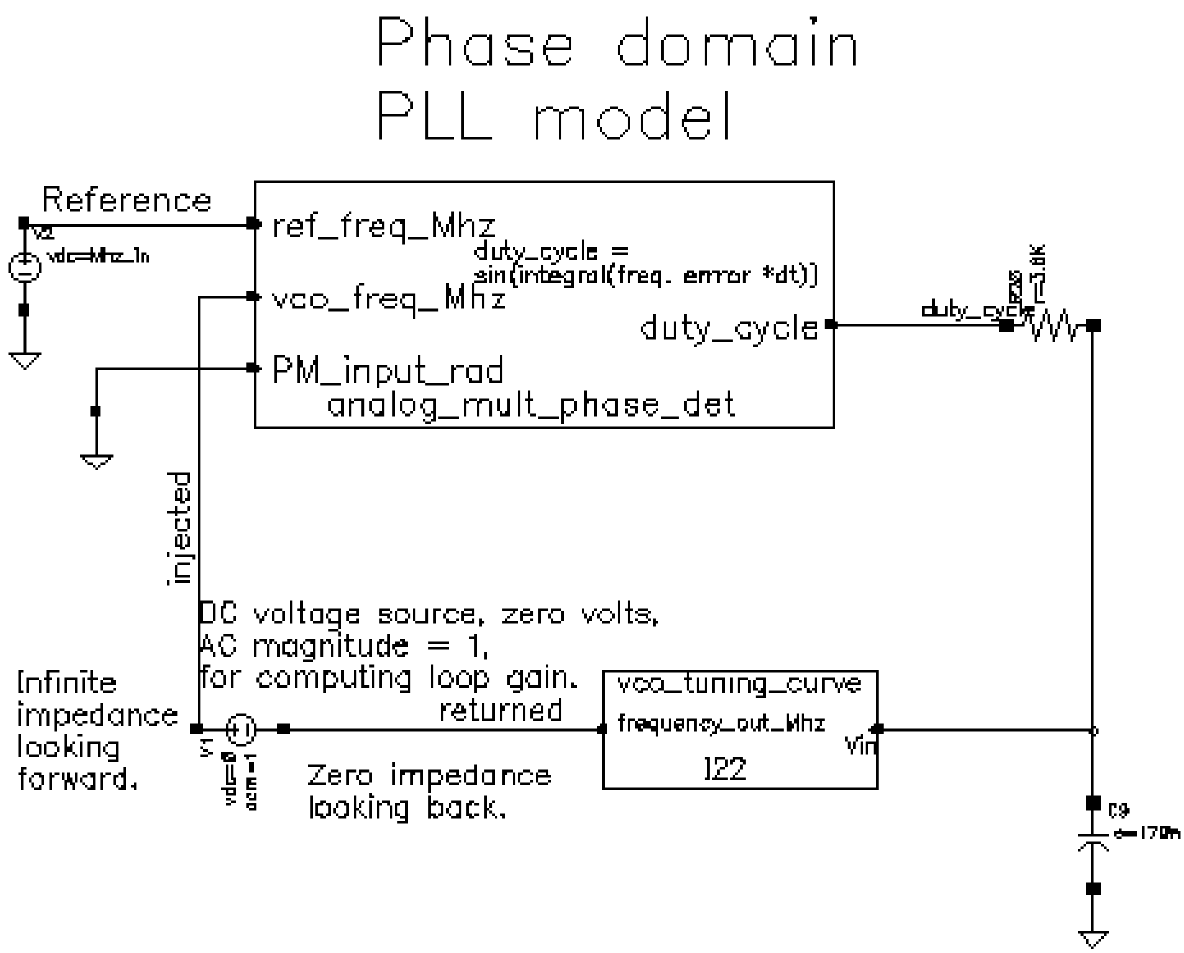
Example 2: Loop Gain Measurement

Spectre RF cannot perform a useful AC analysis on a voltage-domain model because by design, a voltage-domain PLL model has no DC operating point. However, because Spectre RF linearizes phase-domain models about phase error, and phase error is a meaningful DC quantity, subsequent AC analyses are valid.

This example describes how to compute loop gain with a phase-domain model. Figure 3-8 on page 406 shows an analog design environment version of the model shown in Figure 3-4 on page 402. The circuit used to measure loop gain is *example_loop_gain* in the *pllLib* library.

© 1994-2020
All Rights Reserved.

Figure 3-8 Set Up for Loop Gain Measurement



The phase-domain model in Figure 3-8, *example_loop_gain*, includes a voltage source inserted after the VCO. The DC voltage is zero volts, and the AC magnitude is 1 volt. The new voltage source inserts a test signal without changing the DC operating point. You must insert this source at a point where the impedance looking back is much smaller than the impedance looking forward. The accuracy of the resulting loop gain computation depends on how well this condition is met.

Use the following procedure to compute the loop gain.

Open the example_loop_gain Schematic

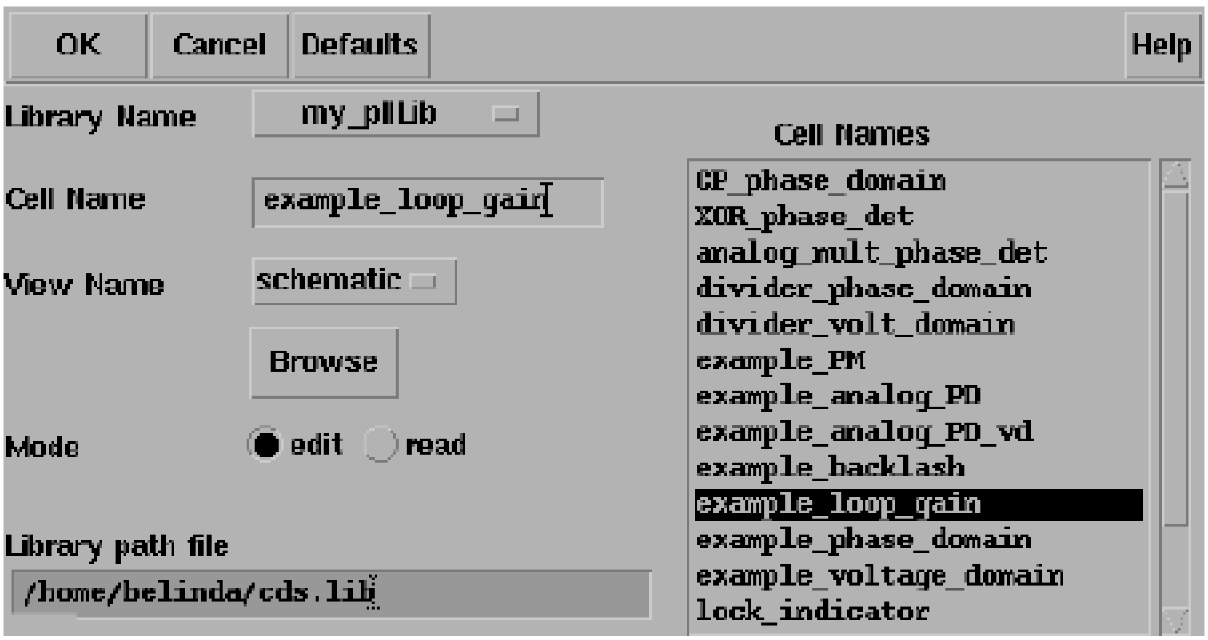
1. In the CIW, choose *File – Open*.

   The Open File form appears.

2. In the Open File form, choose *my_pllLib* in the *Library Name* cyclic field. Choose the editable copy of the pllLib library you created. (You can create an editable copy of the *pllLib* in the same way as is described in the *Spectre Circuit Simulator and Accelerated Parallel Simulator RF Analysis in ADE Explorer Workshop*.

3. Choose *example_loop_gain* in the *Cell Names* list box.

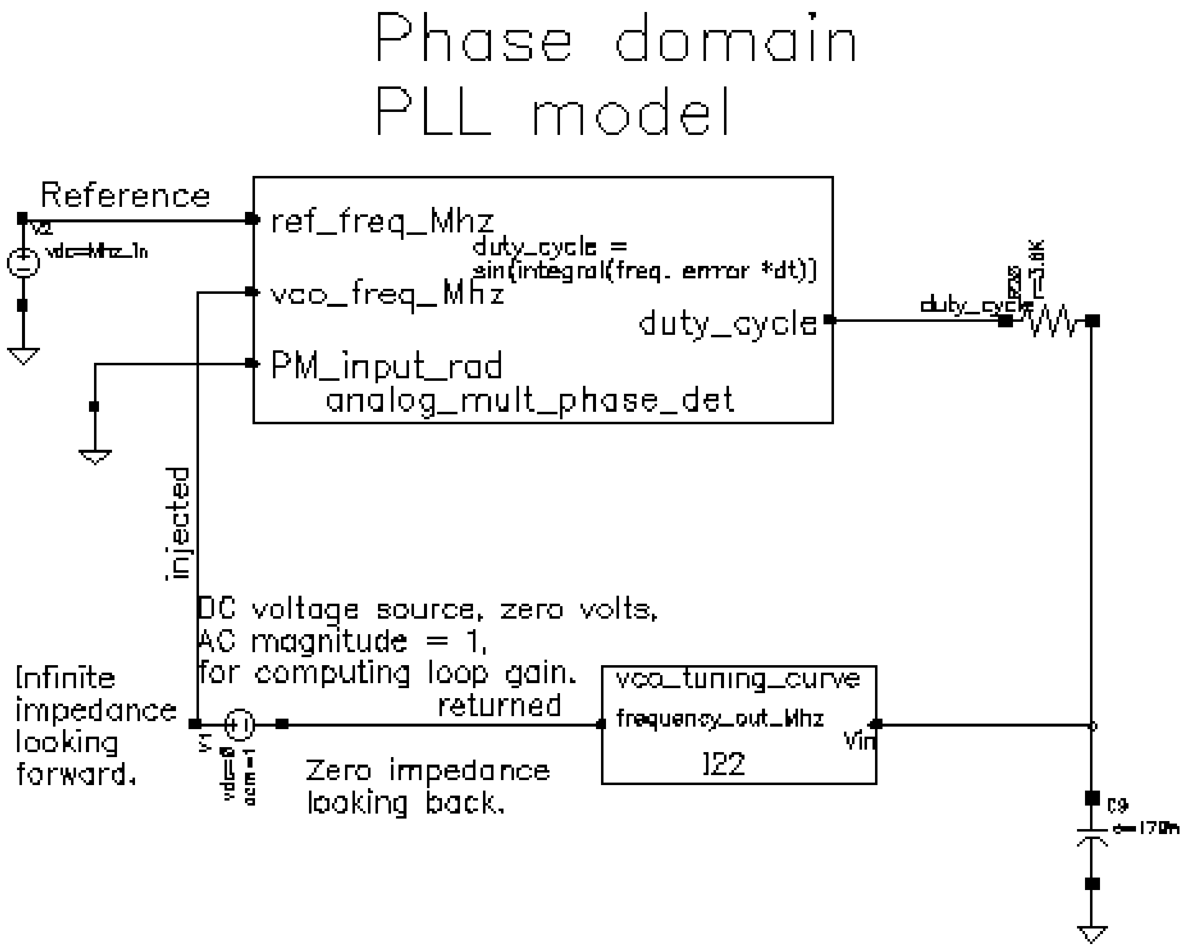
   The completed Open File form appears like the one below.



4. Click *OK*.

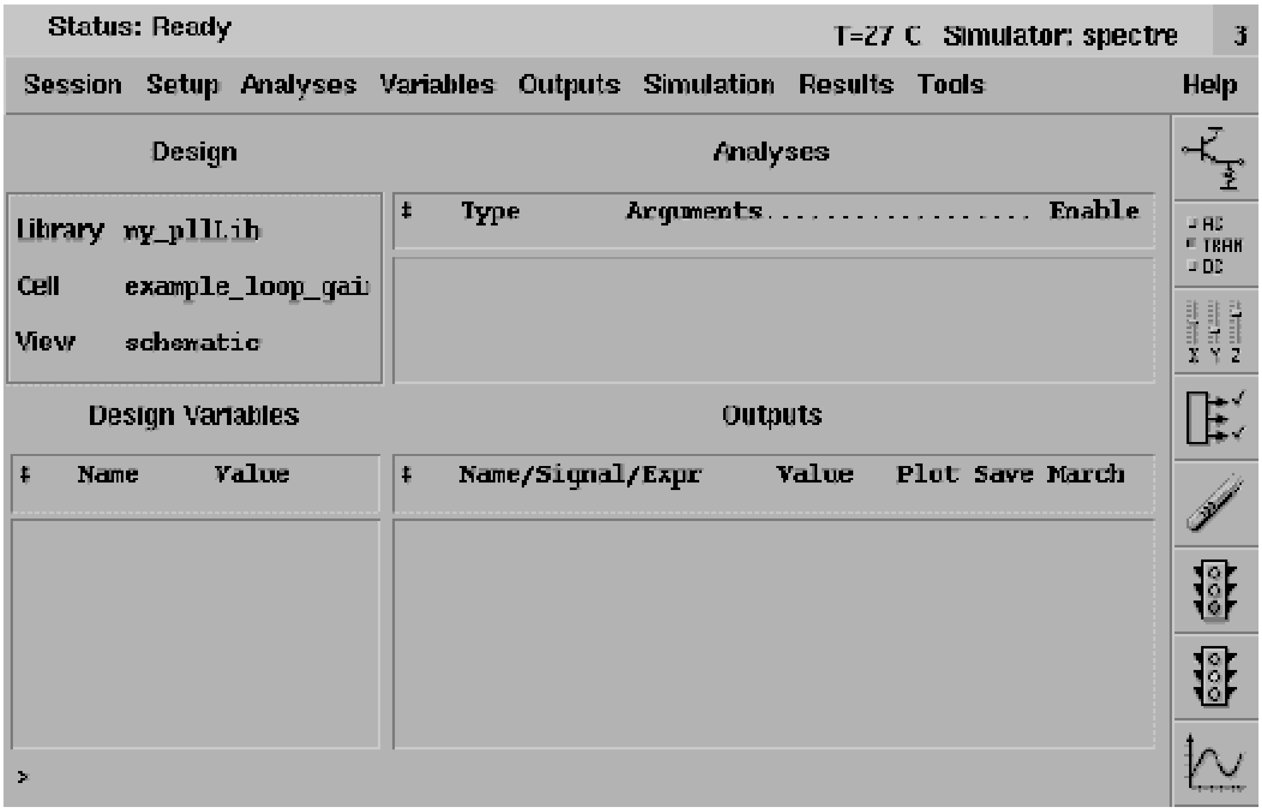
© 1994-2020
All Rights Reserved.

The Schematic window for the *example_loop_gain* circuit appears.



5. In the Schematic window, choose *Tools– Analog Environment*.

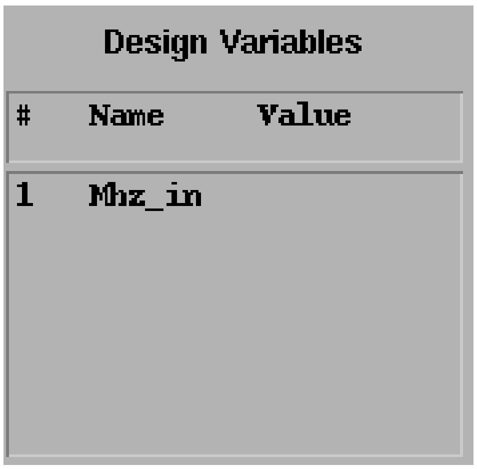
The ADE window opens. This window is also called the Cadence® Analog Circuit Design Environment.



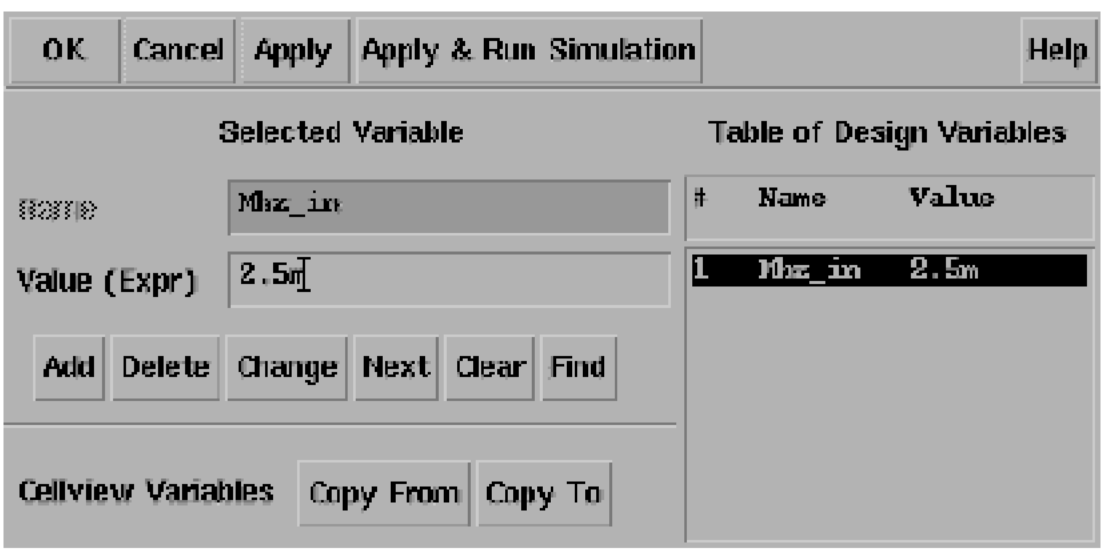
© 1994-2020
All Rights Reserved.

Setting Up the Design Variables

1. In the ADE window, choose *Variables – Copy From Cellview* to copy variables from the schematic to the ADE window. *Mhz_in* displays in the *Design Variables* area of the ADE window.



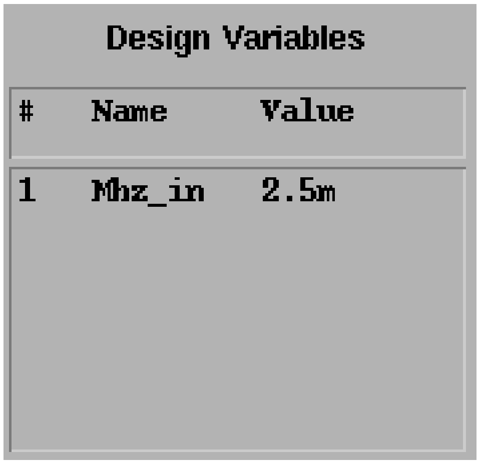
2. In the ADE window, choose *Variables – Edit*, to provide a value for the *Mhz_in* variable.
3. The Editing Design Variables form appears.



4. In the *Value (Expr)* field, type `2.5m` for the value of `Mhz_in` and click *Change*.
5. In the Editing Design Variables form, click *OK*.

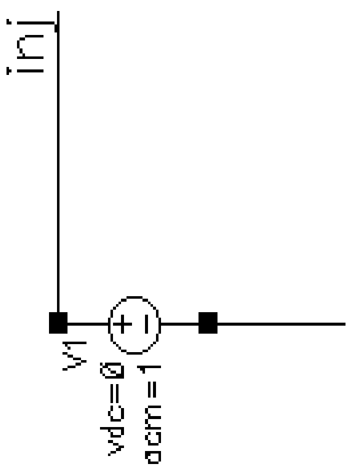
© 1994-2020 All Rights Reserved.

6. The new value for *Mhz_in* displays in the *Design Variables* area of the ADE window.



Setting Up the AC and DC Simulations

Set up both AC and DC analyses. The zero-voltage *vdc* source must be the only source with a non-zero AC magnitude.



When you set up the DC analysis, save the DC data so you can annotate the schematic with DC node voltages.

1. In the ADE window, choose *Analysis - Disable* to disable any analyses you ran previously. (Check the ADE window to verify whether or not any analysis is enabled.)

2. In the ADE window, choose *Analysis - Choose* to display the Choosing Analyses form.

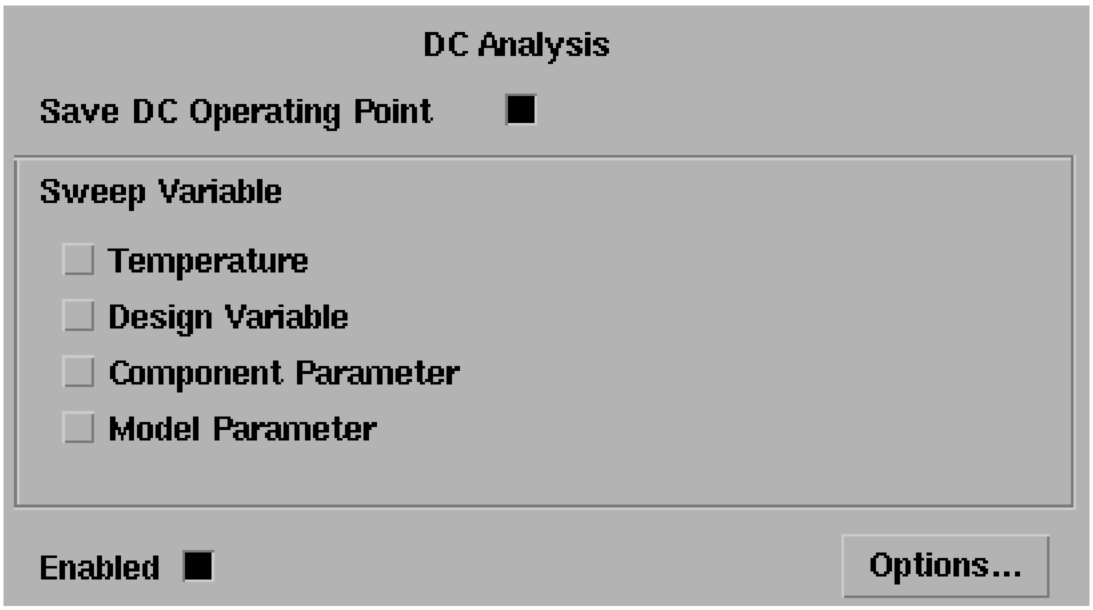
© 1994-2020
All Rights Reserved.

3. Click *dc* to set up the DC analysis.

4. In the DC Analysis area

  a. Highlight *Save DC Operating Point*.

  b. Highlight *Enabled*.



5. Click *ac* to set up the AC analysis.

6. In the AC Analysis area

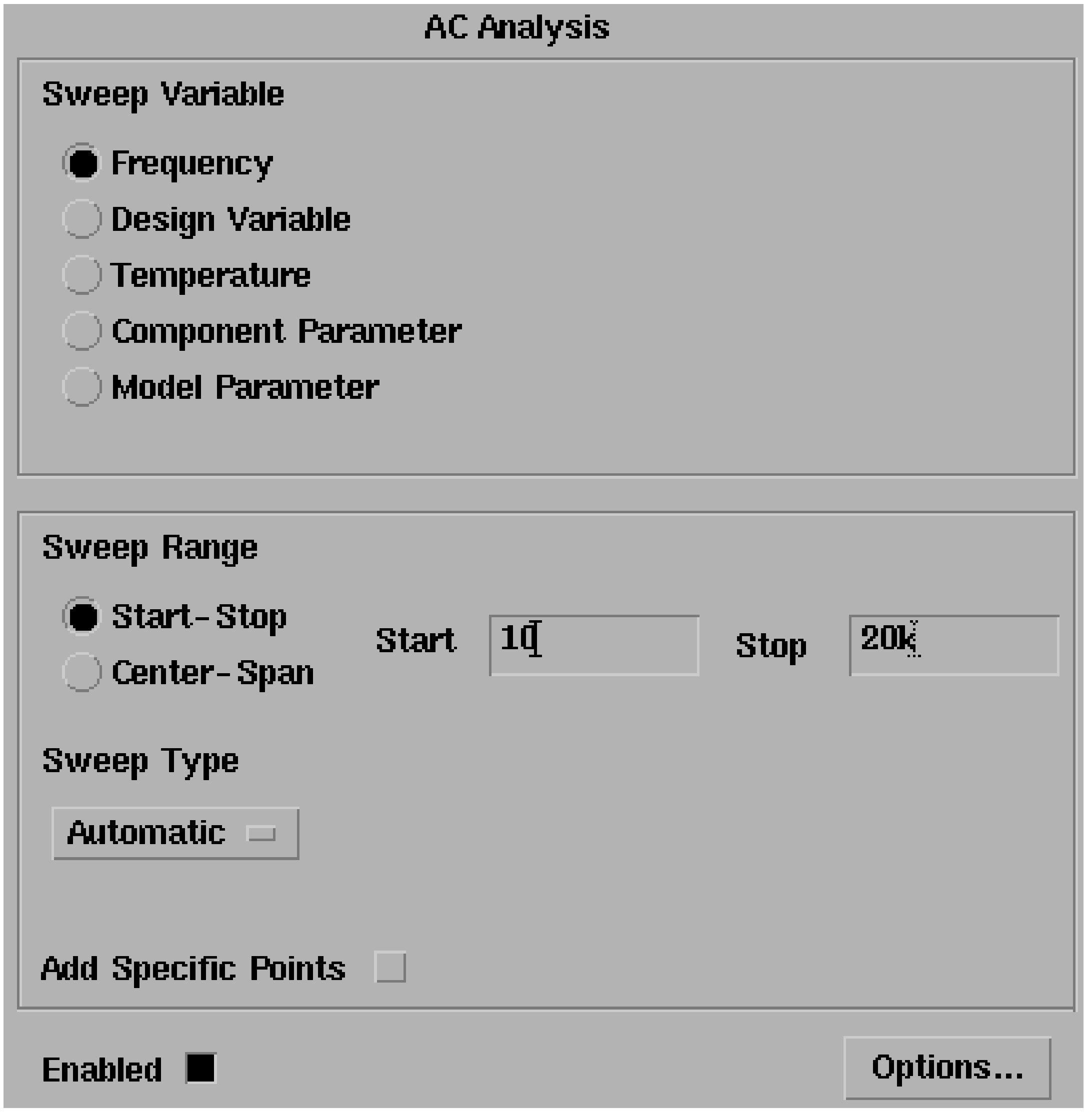
  c. Highlight *Frequency* for *Sweep Variable.*

  d. Highlight *Start - Stop* for *Sweep Range*. Type `10` in the *Start* field and `20k` in the *Stop* field.

  e. Select *Automatic* in the *Sweep Type* cyclic field.

© 1994-2020
All Rights Reserved.

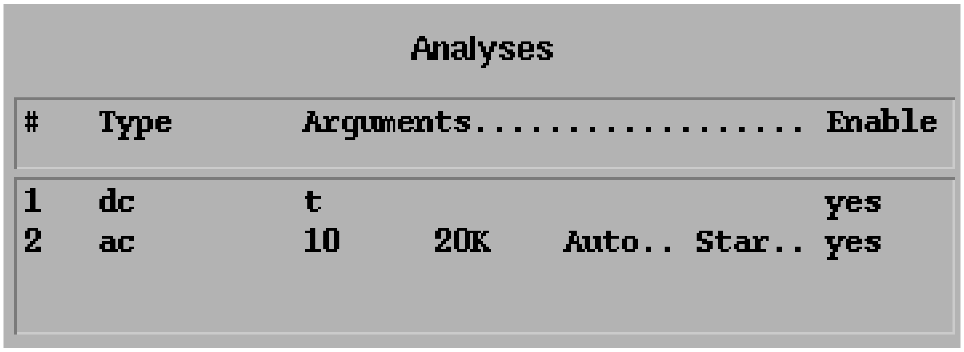
f. Highlight *Enabled*.



7. Click *OK* in the Choosing Analyses form.

© 1994-2020
All Rights Reserved.

8. Both analyses are displayed in the ADE window.



Analyses

#	Type	Arguments..................				Enable
1	dc	t				yes
2	ac	10	20K	Auto..	Star..	yes

Run the Analyses

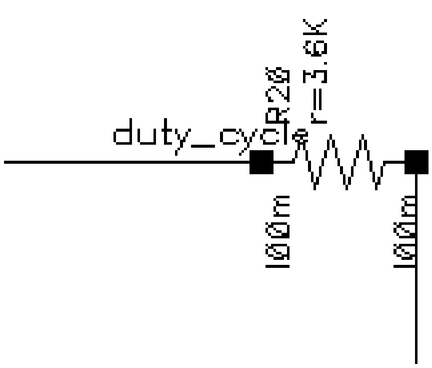
1. To run the analyses, choose *Simulation – Netlist and Run* in the ADE window.

   The output log file appears and displays information about the simulation as it runs.

   Look in the CIW for a message that says the simulation completed successfully.

Displaying the DC Voltages on the Schematic Nodes

➤ In the ADE window, choose *Results – Annotate – DC Node Voltages* to display the node voltages on the schematic. The DC operating point for the net called *duty_cycle* must remain between `-1` volt and `1` volt.

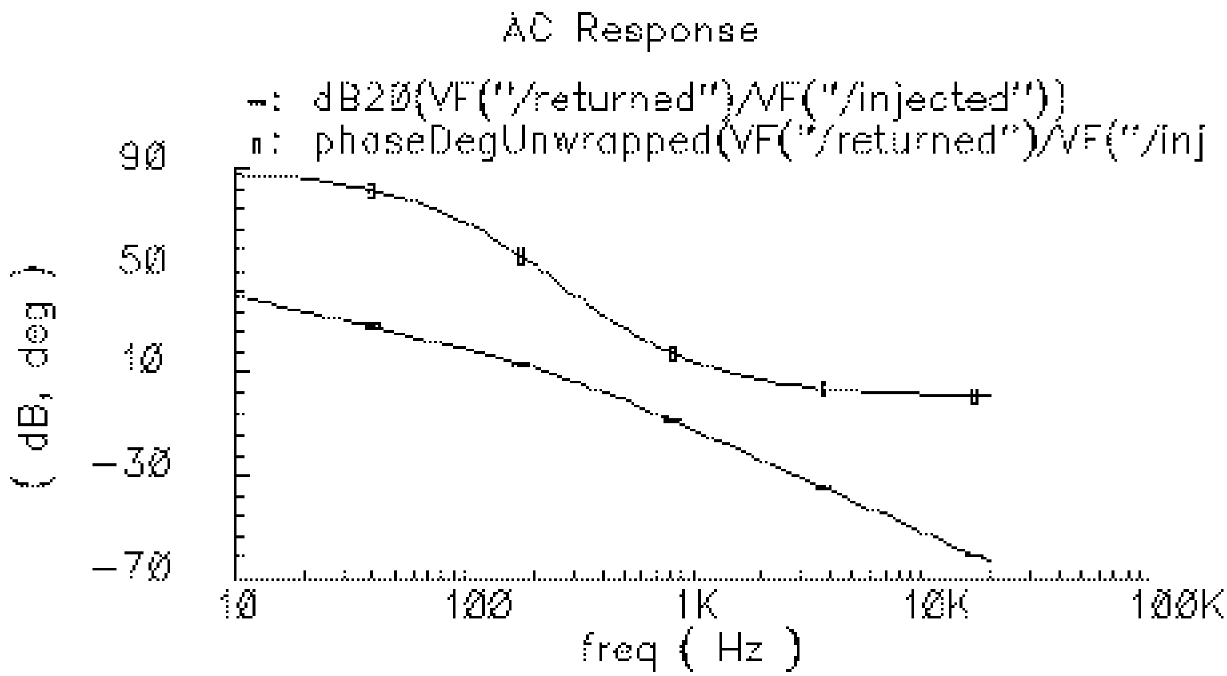


If the operating point falls outside the interval [-1, 1] volt, the loop is not locked and the AC analysis is invalid.

All Rights Reserved.

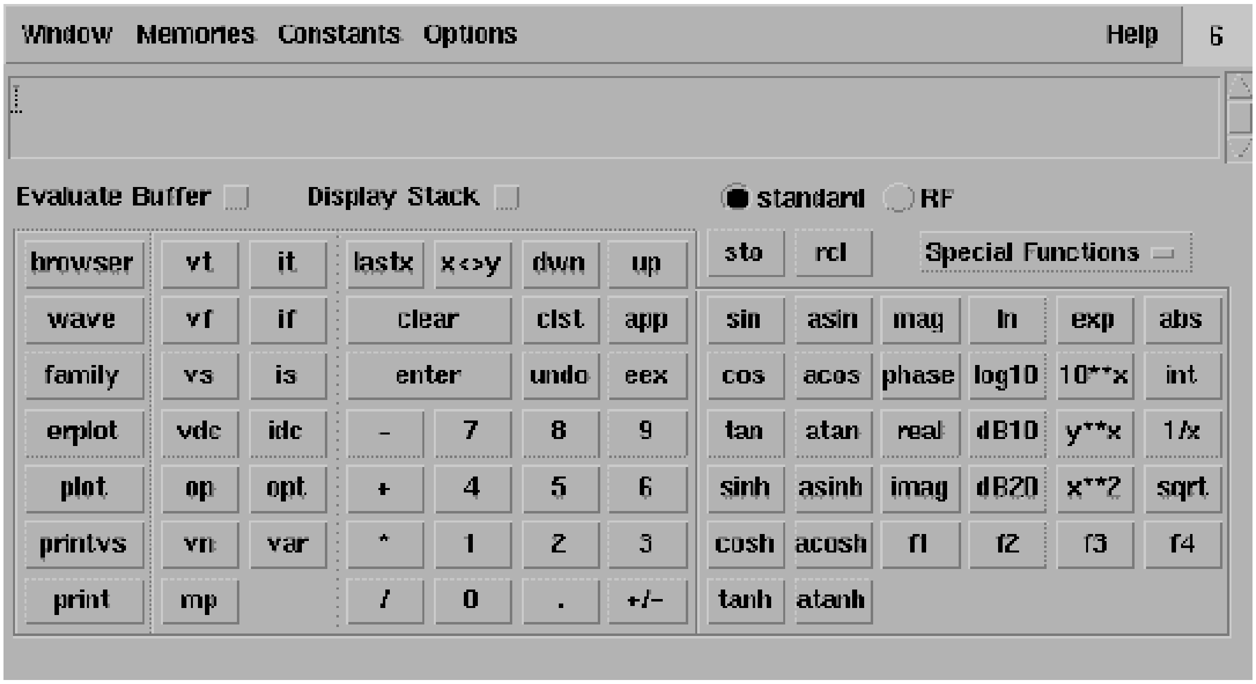
AC Response as Gain and Phase

1. In the ADE window, choose *Results – Direct Plot – AC Gain & Phase* and follow the prompts at the bottom of the Schematic window.
2. *Select first point*—Select the net labeled *returned* in the schematic.
3. *Select second point*—Select the net labeled *injected* in the schematic.
4. The waveform window displays two curves.
5. The top curve plots phase.
6. The bottom curve plots gain.



© 1994-2020
All Rights Reserved.

7. In the waveform window, choose *Tools – Calculator* to open the calculator.



8. In the calculator, click the *wave* button (on the left).

9. Then, in the waveform window, select the phase curve (on the top).

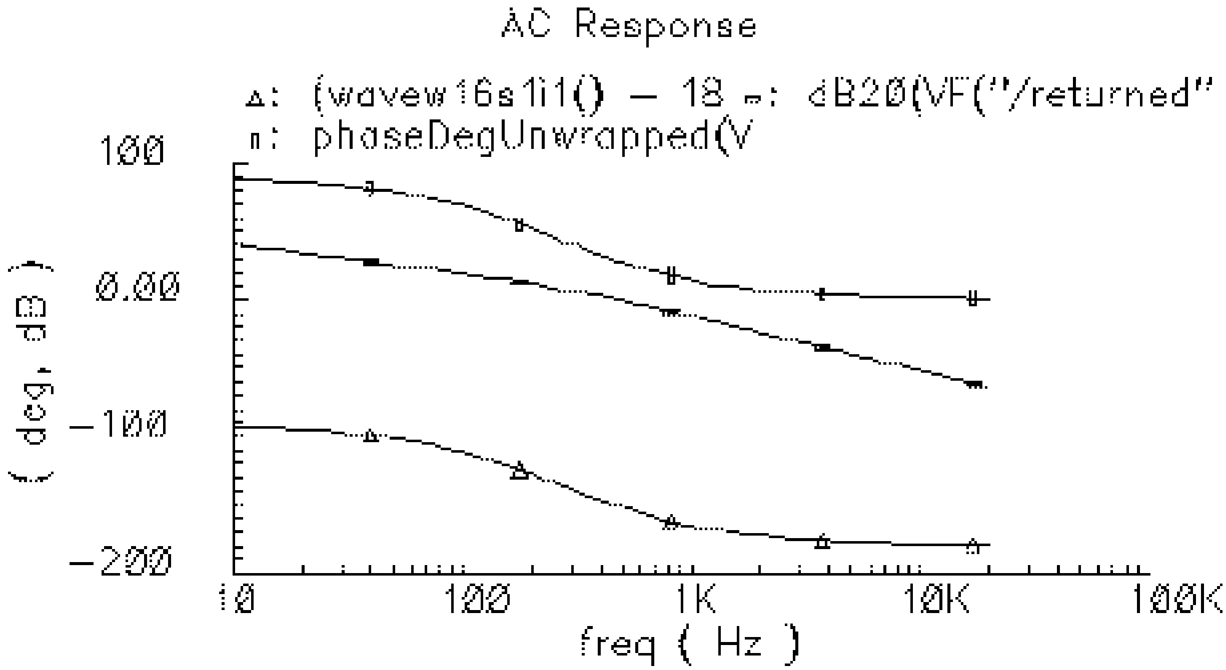
10. In the calculator, to perform the calculations algebraically, choose *Options – Set Algebraic*.

11. Subtract 180 from the phase waveform—In the calculator, click the subtraction symbol (-) followed by the numbers *180*.

12. The calculator buffer should look similar to the following

© 1994-2020
All Rights Reserved.

13. Click the *plot* button to plot the calculated waveform.



14. To remove the original phase curve from the waveform window, in the waveform window

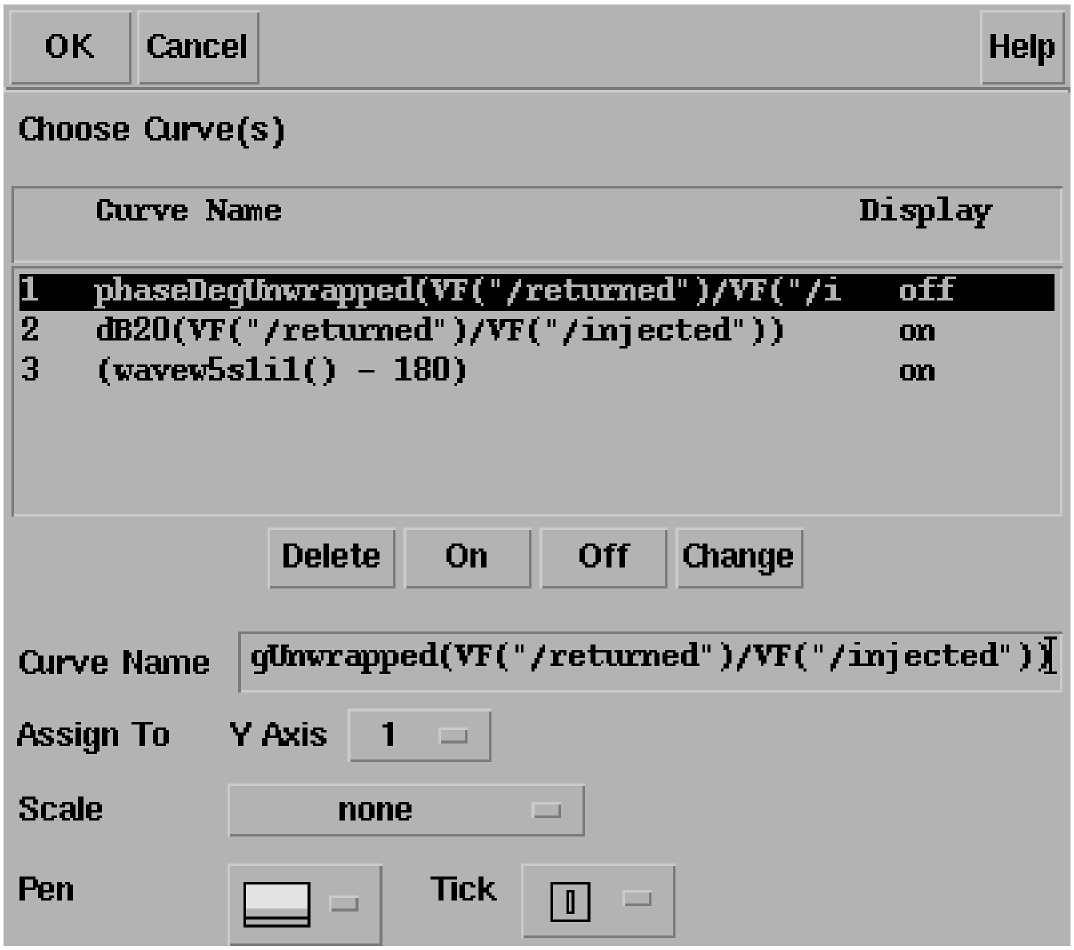
a. Choose *Curves – Edit* to display the Curves form.

b. In the Choose Curves list box, highlight the original phase curve.

c. Click *Off*.

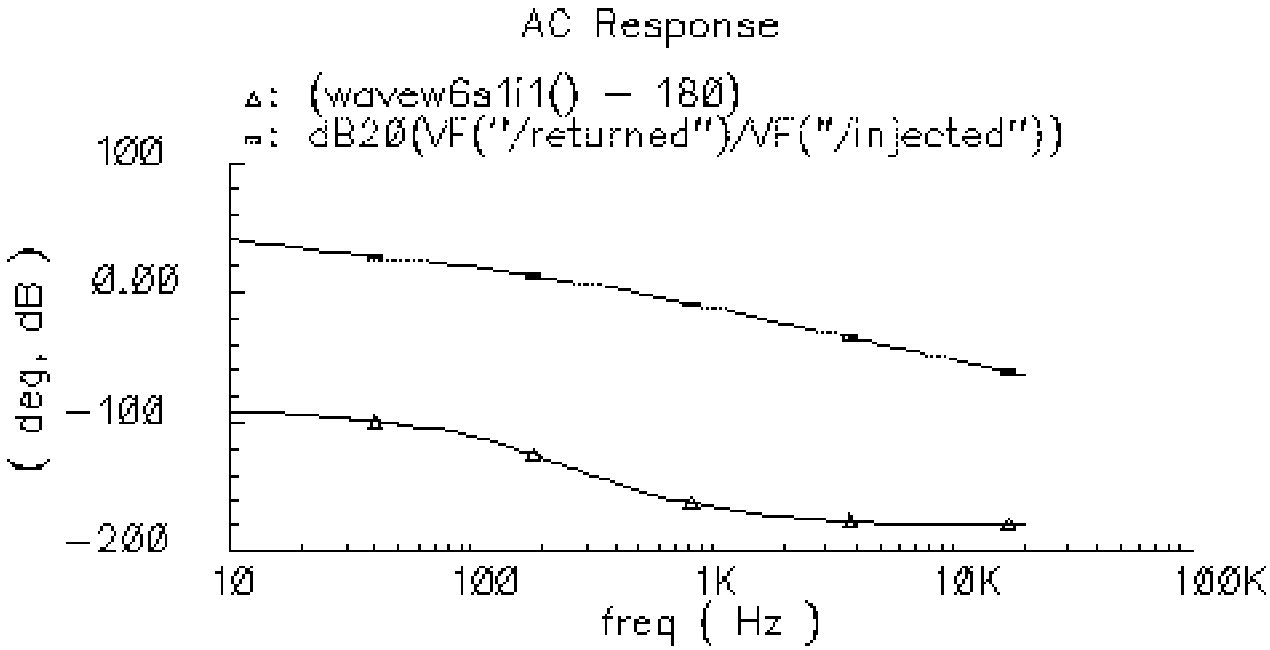
© 1994-2020 All Rights Reserved.

d. The Curves form looks similar to the following.



© 1994-2020 All Rights Reserved.

The original phase waveform is no longer displayed in the waveform window.



15. To create the Bode plot

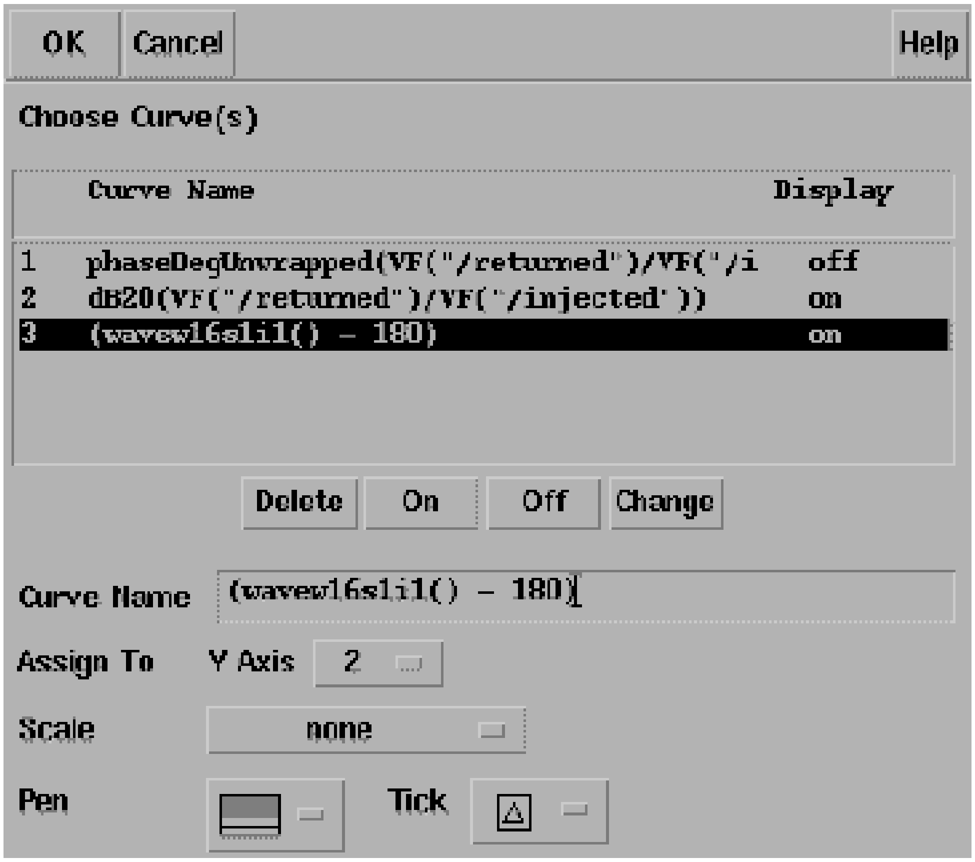
a. If necessary, in the waveform window, choose *Curves – Edit* to display the Curves form.

b. In the Curves form, select the shifted (by 180 deg) phase curve

c. In the Curves form *Assign to Y Axis* cyclic field, select *2*.

© 1994-2020
All Rights Reserved.

d. The Curves form looks similar to the following.



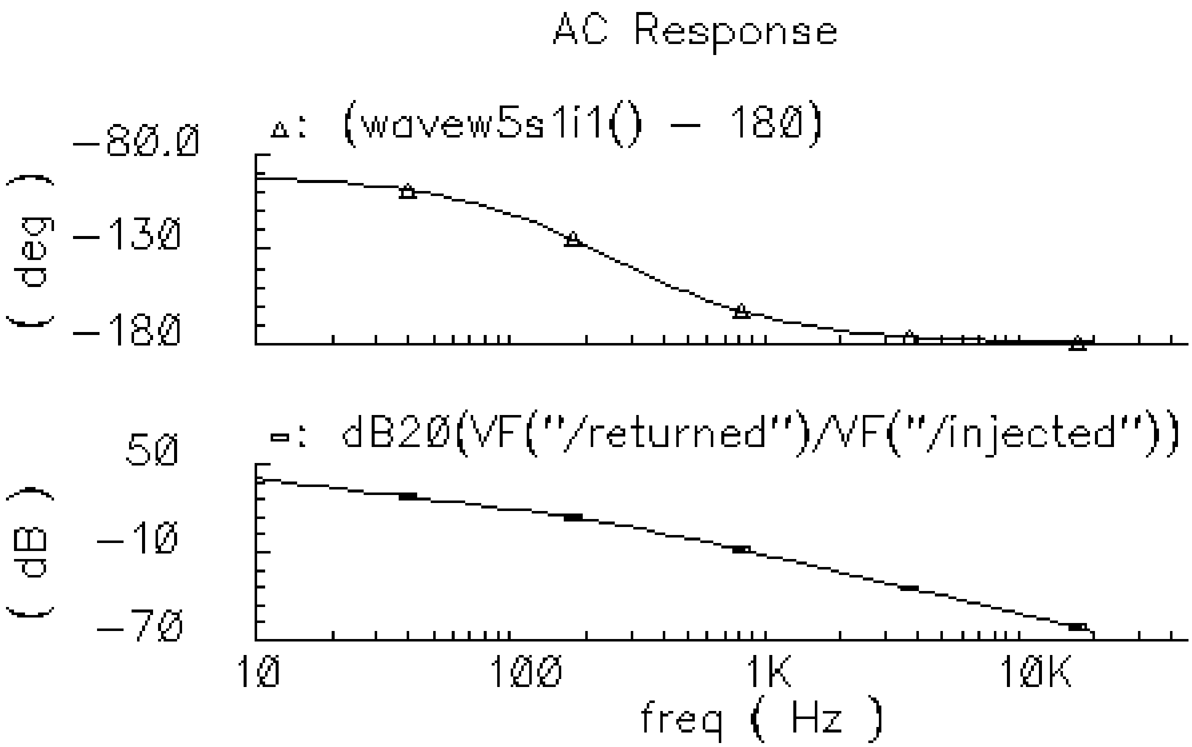
16. Click *OK* in the Edit Curves form.

17. In the waveform window, choose *Axes – To Strip* to change the display as follows.

This produces a Bode plot (magnitude and phase) of the loop gain in the waveform window Shown in Figure 3-9.

© 1994-2020 All Rights Reserved.

Figure 3-9 Bode Plots, Magnitude and Phase of Loop Gain



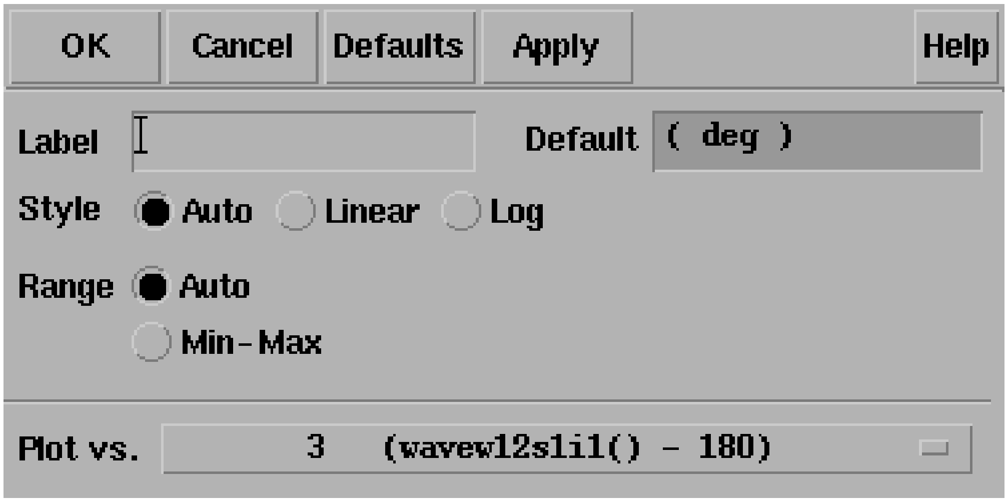
18. To generate a Nichols chart (dB versus degrees) from which you can pick off phase and gain margins,

a. In the waveform window, choose *Axes – X Axes*

b. In the *Plot vs. cyclic* field select the phase curve you created that is 180 degrees out of phase.

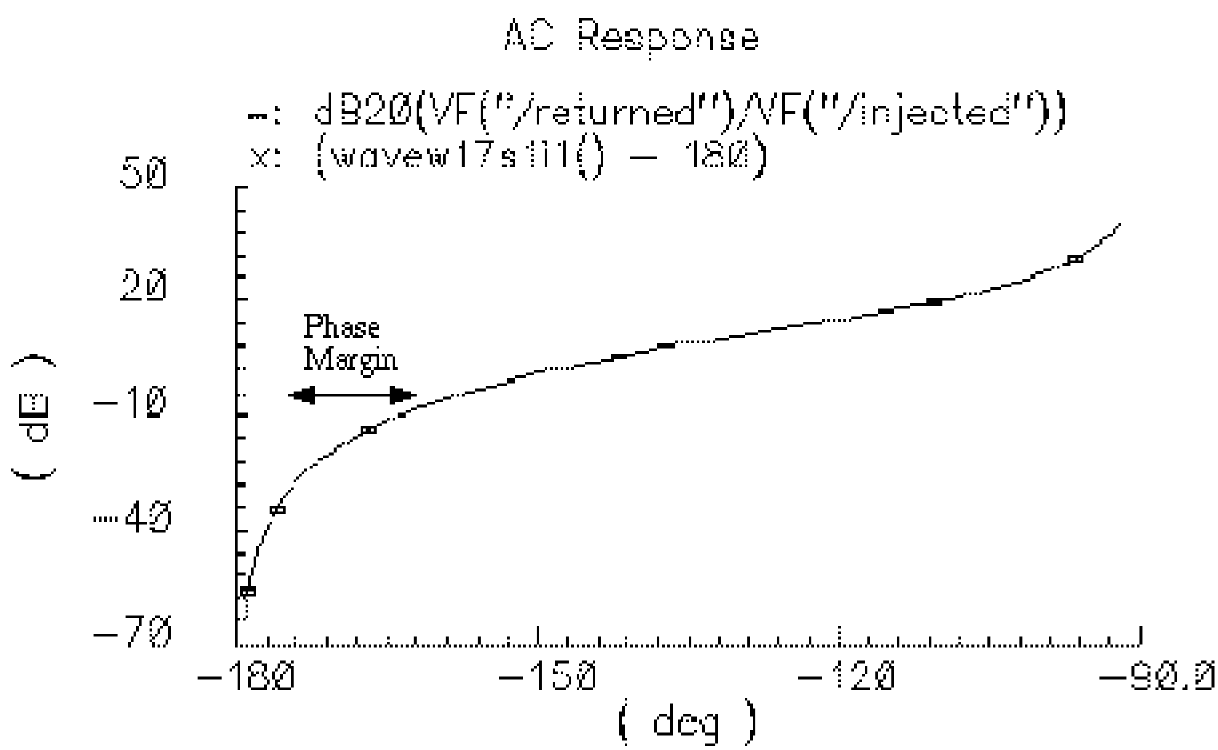
© 1994-2020
All Rights Reserved.

c. Click *OK*.



You now have a Nichols chart like the one in Figure 3-10. The phase margin is 30 degrees.

Figure 3-10 Nichols Chart of Loop Gain



© 1994-2020
All Rights Reserved.

To compute phase margin directly do the following:

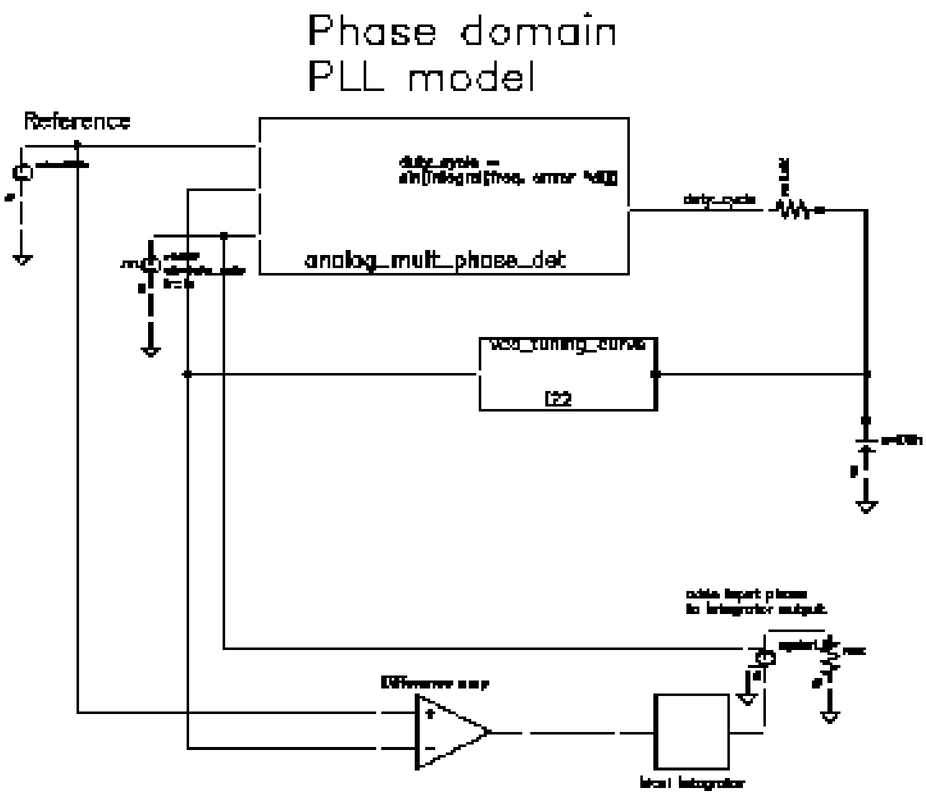
1. In the calculator, click *vf*.
2. In the Schematic window, click first on the return node and then on the injected node.
3. In the calculator, click the *divide* button.
4. In the calculator Special Functions menu, choose *phase margin* followed by *print*.
5. Add 180 degrees to the expression in the calculator then choose *print* from the Special Functions menu.
6. The Results Display Window displays the phase margin.

Example 3: PM Input

The circuit used to test for PM input is *example_PM* in the *pllLib* library. The PM (phase modulation) input pin is useful if the PLL is used as a modulator or demodulator, but it also provides a convenient place to perturb the PLL to assess large signal stability. Figure 3-11 on page 424 shows a test circuit for such a stability check.

© 1994-2020
All Rights Reserved.

Figure 3-11 Stability Check Using the PM Input

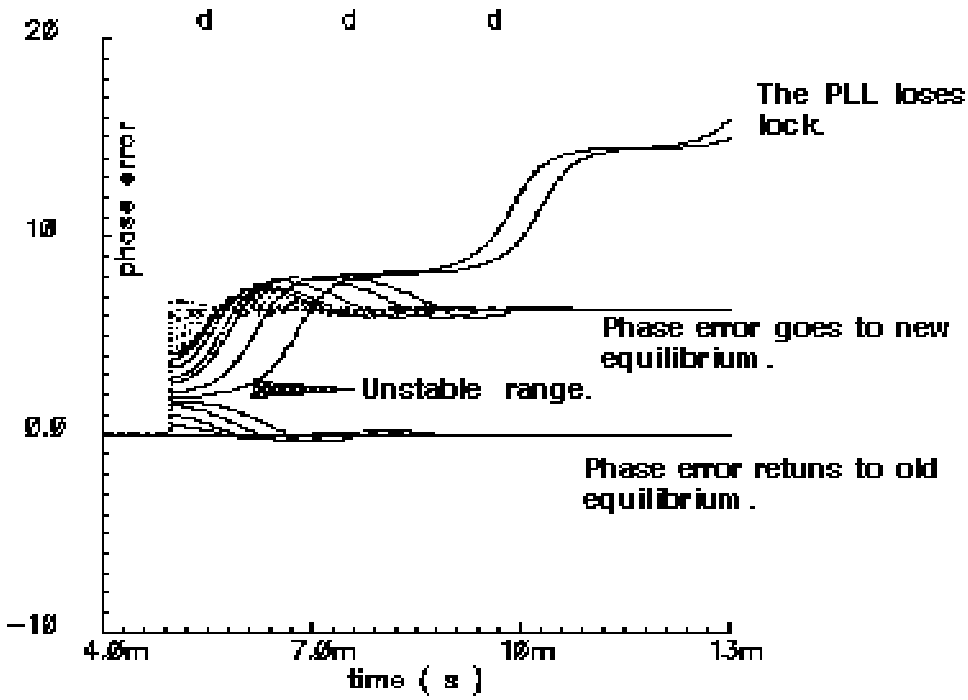


This PLL is very simple but yet different from the previous example. It was modified to produce more interesting results. The lower circuitry requires explanation. The difference amplifier computes frequency error and converts it from Mhz to rad/sec. The integrator computes VCO and reference contributions to phase error. The voltage-controlled-voltage-source at the end adds the input phase stimulus to compute total phase error. The difference amplifier and integrator are from the *ahdLib*.

The input phase is a delayed step. The delay makes the initial phase error easy to read. A parametric analysis on the phase error's step response with respect to the size of the input phase step reveals some interesting behavior. Figure 3-12 on page 425 shows the family of phase error step responses produced by the parametric analysis. The external integrator is intentionally not a circular integrator like the one inside the phase detector model. For large and small steps in input phase, the PLL settles into equilibrium, possibly a new one. However, a narrow intermediate range of steps puts the PLL into an unstable mode. The references examine this behavior in mathematical detail [1,4,6,10]. This example shows one way to assess large-signal stability and to demonstrate that the phase-domain models capture the major non-linear mechanisms.

© 1994-2020 All Rights Reserved.

Figure 3-12 Phase Error Response to Step PM Input

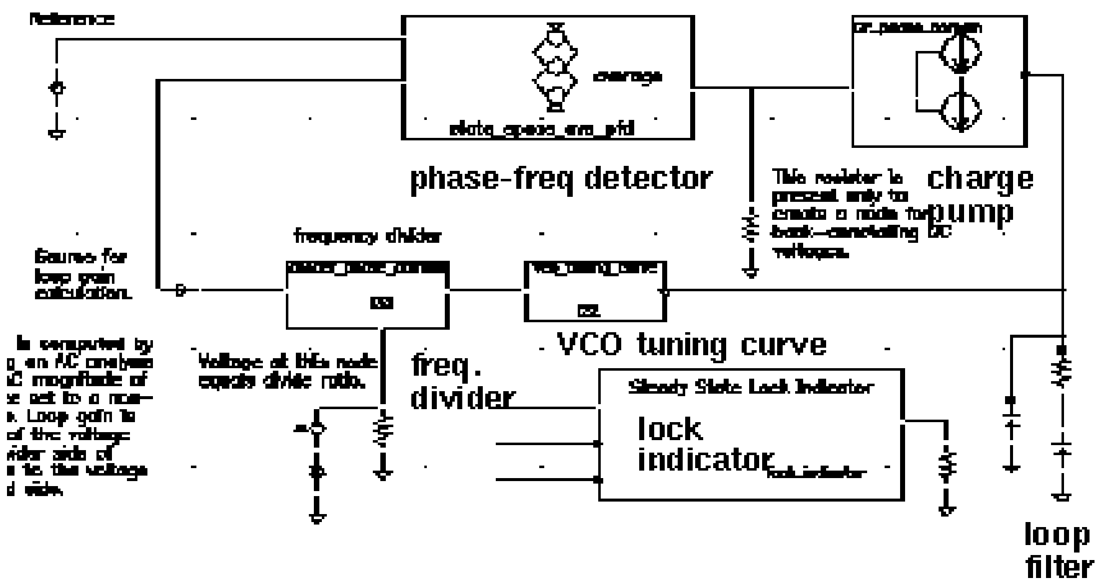


Modeling a PFD-Based PLL

Figure 3-13 on page 426 shows a block diagram of a typical PLL with a phase-frequency detector. This section describes how to specify each component in Figure 3-13 and briefly explains what each model does.

© 1994-2020
All Rights Reserved.

Figure 3-13 PLL Block Diagram



VCO

The VCO is modeled by its tuning curve. The tuning curve characterizes the relationship between the input voltage and the output frequency. The input to the VCO model is the loop filter output voltage, also called the VCO control voltage. The VCO output is a voltage representing the VCO's instantaneous frequency in Mhz. Therefore, when the VCO operates at 2 Mhz, the model output is 2 Volts.

The VCO tuning curve is generally nonlinear and can be specified in one of two ways:

- With the coefficients of a fourth order polynomial
- With a look-up table

Polynomial tuning curve: The input voltage is internally clamped to the nearest end point if it moves outside the interval [*min-vco-input-voltage*, *max-vco-input-voltage*]. Although the input voltage may fall outside the interval, the output behaves as though the input voltage value is at the end points. Within the interval, the output is a fourth order polynomial in the quantity, V_{input} minus the free running voltage. When the input voltage equals the free running voltage, the output frequency equals the free running frequency. The scale factor scales the entire polynomial and has a default value of 1. The scale factor is useful in converting data in Khz/volt, for example, to the required Mhz/volt. The parameters are the coefficients of the polynomial.

© 1994-2020 All Rights Reserved.

Table look-up tuning curve: The two parameters are the *scale factor* and the *path to the look-up data*. The look-up model linearly interpolates between data points and linearly extrapolates outside the data interval. The data format is two columns of data delimited by spaces. There is no header, and there are no extra lines at the end. The first column is input voltage. The second column is output frequency. The path to the data can be absolute or relative to the netlist. The netlist is usually stored at

```
<home>/simulation/ckt_name/spectre/schematic/netlist/input.scs
```

but you can choose a different location.

If the data is at

```
<home>/data/table
```

and the netlist is as shown above, the relative path is

```
../../../../../data/table.
Frequency Divider
```

The frequency divider is essentially a simple gain element. It takes an input voltage that represents frequency in Mhz, and then scales it by the divide ratio to generate an output voltage that represents the divided frequency. The divide ratio is numerically equal to the voltage on the control pin. If the divide ratio drops below 0.001, the model assigns it to 0.001 and issues a warning. This assignment prevents division by zero during simulation.

Charge Pump

The charge pump transforms the duty cycle into the expected average current sourced or sunk by the charge pump. You define the maximum source and sink currents, and they can be different from each other. If the charge pump output voltage exceeds the rails you define, the output voltage is clamped to the rail through a 0.001 Ohm resistance. The other parameters are the leakage resistance and open circuit voltage. These last two parameters specify the Thevenin equivalent circuit of a leakage path. The leakage path can source or sink current depending on the open circuit voltage.

Loop Filter

The loop filter is entered component-for-component.

© 1994-2020 All Rights Reserved.

State-Space Averaged PFD (Phase-Domain Phase-Frequency Detector Model)

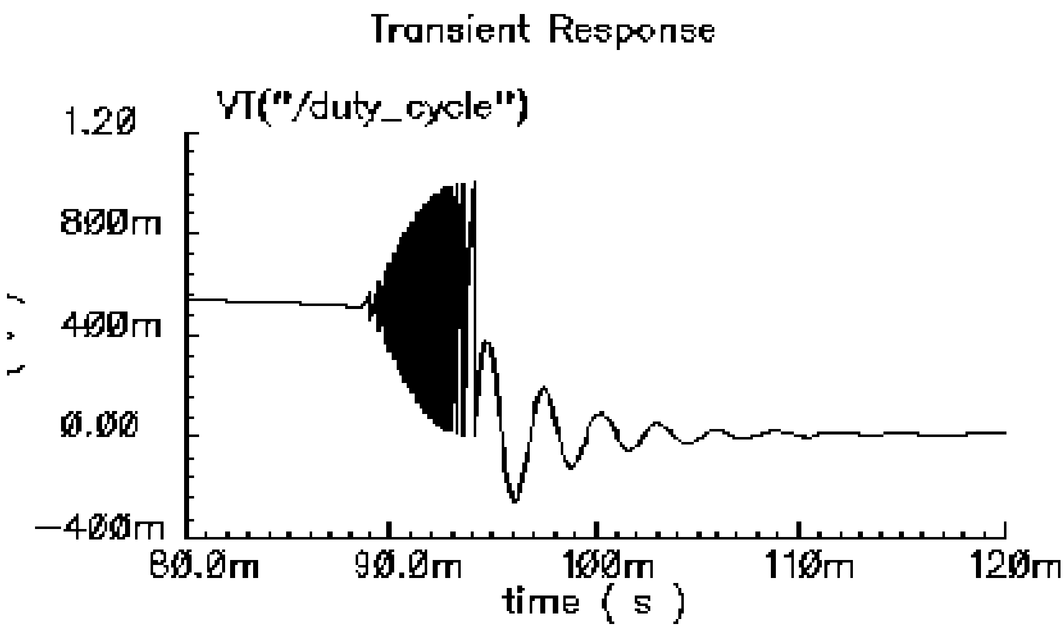
The phase-frequency detector (PFD) model approximates average behavior of a digital, three-state, phase frequency detector. This is the most complicated model in the PLL library. The term *state-space averaged* is borrowed from the power electronics field [14]. The charge pump currents for the three PFD states are averaged together with a duty cycle much like voltages are averaged together with a duty cycle in a switch-mode power supply model. The PFD inputs are voltages representing the reference and divider output frequencies in Mhz. The output is a voltage that when multiplied by the maximum charge pump current, numerically equals the average charge-pump output current. The PFD output is a duty cycle. When the frequency error is large, the duty cycle is a smooth waveform directly related to the normalized frequency error [1,4]. When the frequency error is small, an integrator inside the PFD model converts frequency error to phase error, and the duty cycle is proportional to the phase error. The duty cycle starts jumping to zero (or resets) as it changes from frequency-mode to phase-mode.

As phase error enters a deadband determined by the minimum-on-time parameter and reference frequency, the model computes a duty cycle pulse with magnitude one and duration equal to the minimum-pulse-width. After the pulse expires, the duty cycle drops to zero until the phase error exits the deadband. As the phase error exits the deadband, the duty cycle increases to a non-zero value. The deadband and fixed-width unity pulse simulate what some texts call *backlash* [8].

The PFD model has two parameters. The first is a numerical option that controls the trade-off between execution speed and accuracy. The *speed_vs_accuracy* parameter controls the number of times the internal integrator is reset during the transition from frequency-mode to phase-mode. Too few resets can cause error. Too many resets can needlessly slow run time. The default value of this parameter is 50k. To reduce the number of resets in a slow PLL, and thereby reduce run time, increase the *speed_vs_accuracy* parameter to 70k or 100k. To increase the number of resets in fast PLLs, and thereby increase the accuracy, reduce the *speed_vs_accuracy* parameter to 10k or 20k. A reasonable setting for the *speed_vs_accuracy* parameter produces a duty cycle step response that resets approximately to zero at least 3-10 times before entering the final transient. Figure 3-14 on page 429 shows reasonable duty cycle step response.

© 1994-2020 All Rights Reserved.

Figure 3-14 Reasonable Duty Cycle Waveform



The other parameter is the *minimum_on_time* which controls the backlash. This is the minimum pulse width the PFD can generate. As the phase error decreases, the pulse width drops discontinuously from the minimum pulse width to zero. This effect creates a deadband in the duty cycle versus phase error curve.

Figure 3-12 on page 425 was generated with the default *minimum_on_time* parameter value of zero µs. The default value of the *minimum_on_time* parameter produces no deadband and no unity pulses. The default deactivates the backlash mechanism.

Figure 3-15 on page 430 was generated with a *minimum_on_time* parameter value of 0.2 µs. Figure 15a illustrates that the pulses only occur as the phase error enters the deadband. Figure 15b shows the limit cycle created by the backlash. The limit cycle is primarily determined by leakage on the loop filter and the minimum pulse width. Some references suggest biasing the duty cycle away from the deadband or loading the filter down to force the limit cycle frequency to a value in which the loop filter attenuates it. The phase-frequency detector model can help quantify the problem and check the solution.

A pulse is not kicked out upon exiting the deadband because that behavior causes convergence problems for Spectre RF. If phase error is entering the deadband, a pulse at that moment pushes phase error in the same direction it was going, into the deadband. If a pulse occurs as phase error exits the deadband, the pulse drives phase error back into the deadband and Spectre RF has trouble figuring out whether phase error should leave the deadband at all. Fortunately, no significant error is introduced by implementing the pulse only when phase error enters the deadband. In a backlash limit cycle, the feedback loop quickly

© 1994-2020 All Rights Reserved.

drives phase error back into the deadband and a pulse occurs on the way in. The error is in the time the feedback loop takes to return phase error to the edge of the deadband and that is usually small when PLL is in a backlash limit cycle.

(The PLL model that generated Figure 3-15 had a center frequency of about 1Mhz and the simulation ran out to 130ms. A voltage-domain model might easily simulate 10 points per carrier cycle. The voltage-domain model would require 1.3 million points to simulate the same amount of action. I did not attempt it. The phase-domain simulation that generated Figure 3-15 ran in a matter of seconds!)

Figure 3-15 Duty Cycle Waveforms With Pulses and Backlash

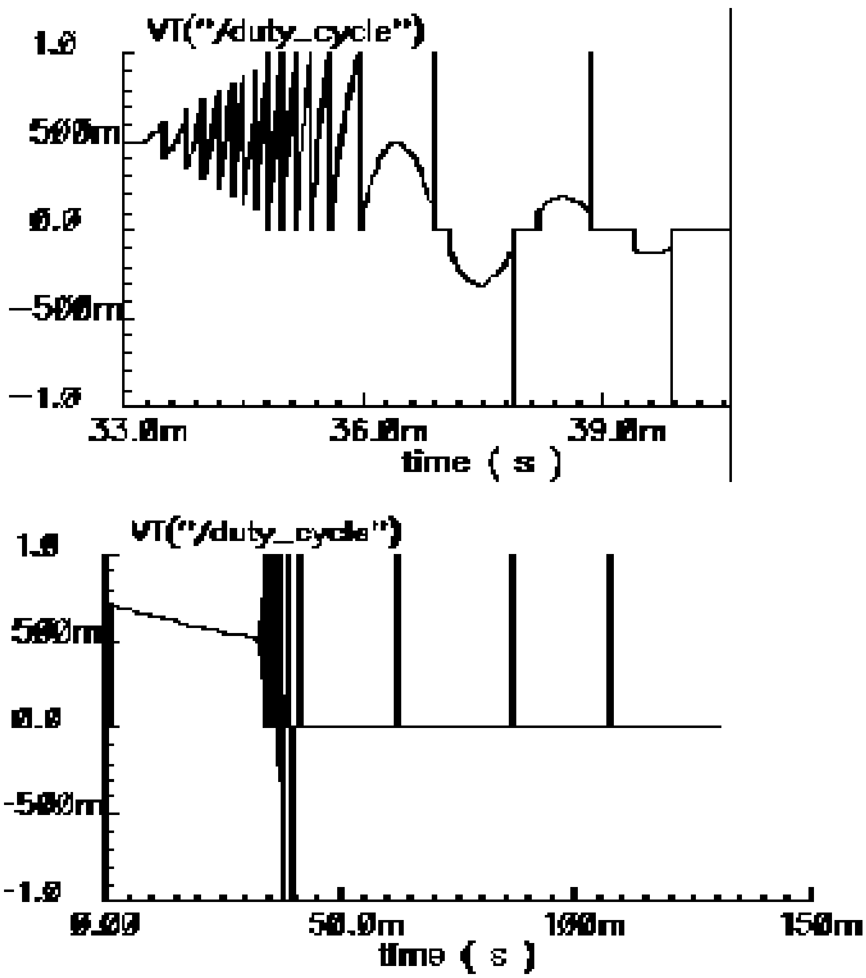


Figure 15a
Duty cycle between
33 ms and 40 ms

Figure 15b
Duty cycle to
130 ms

© 1994-2020
All Rights Reserved.

Lock Indicator

All real components have limited outputs and the limits of any one component can keep the PLL from locking. Just like the simple phase detector model, all of the phase-domain models operate one way for DC analysis and another for transient analysis to prevent DC convergence errors. The lock indicator monitors three signals. In the example, the lock indicator monitors the phase detector output, the VCO control, and the charge pump output. If any of those signals exceeds its limit, the lock indicator output is zero, signifying that the loop is not locked in steady state. If all signals are within their limits, the output is 1 volt, specifying that the loop is locked. The lock indicator is only valid for DC analysis. Use node names to tie the lock indicator inputs to the right nodes and use variables for the component limits. You must specify the units manually twice, once for each component and once for the lock indicator. With variables, the lock indicator parameters are linked to the proper component parameters, and you specify changes in only one place.

Example 4: Modeling Acquisition Transients

The circuit used to model acquisition transients is the *example_phase_domain* in the *pllLib* library. Figure 3-16 on page 432 shows the duty cycle and VCO frequency response to a momentary change in the divider ratio. When the divider ratio changes, the PFD enters the frequency-mode and slews the VCO frequency toward the new value. As the VCO frequency approaches the final value, the PFD model gradually changes from frequency-mode to phase-mode. When the frequency error is small, but still large enough to slew phase error, the duty cycle waveform looks like a sawtooth waveform. The model gradually increases the amplitude of the sawtooth component of the duty cycle, and always maintains the correct average, as frequency error reduces to zero. The final duty cycle transient is the sawtooth that depends mostly on phase error. Figure 3-12 on page 425, modifies the x-axis of the graph to show the duty cycle in the first transition.Figure 3-17 on page 432 modifies it further to show the sawtooth waveform.

© 1994-2020
All Rights Reserved.

Figure 3-16 Response to Momentary Change in Divider Ratio

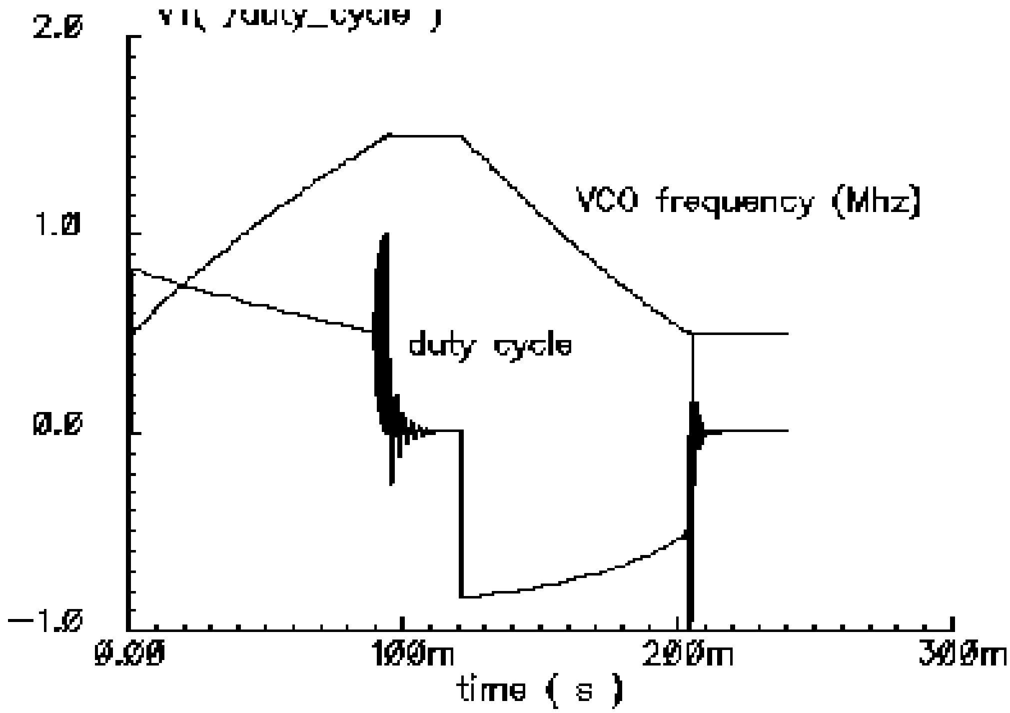
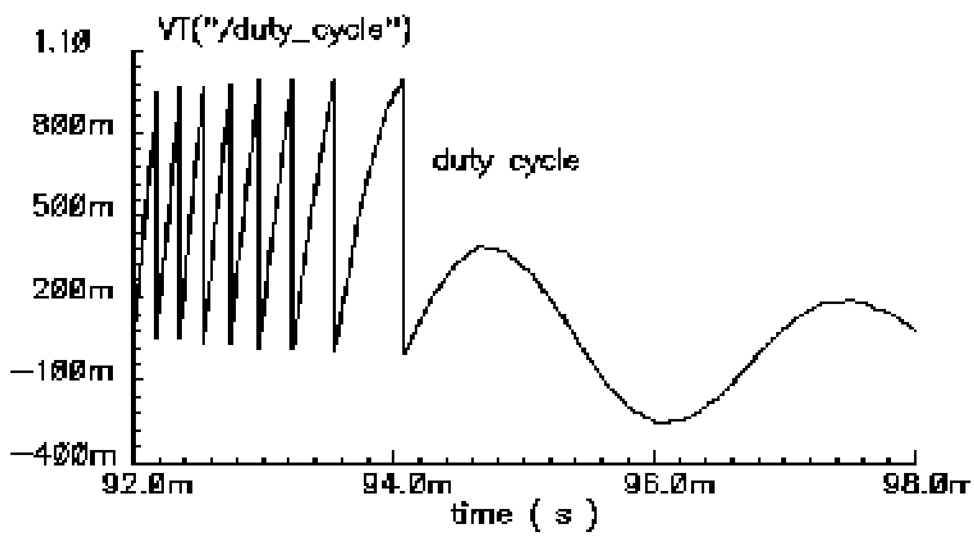


Figure 3-17 Duty Cycle During Transition From Frequency to Phase Mode



© 1994-2020 All Rights Reserved.

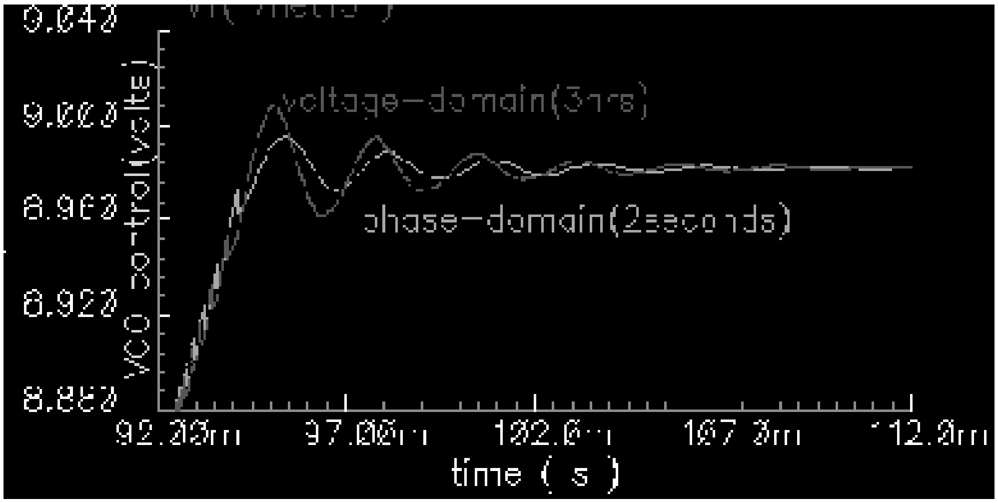
Example 5: Comparison With a Voltage-Domain Model

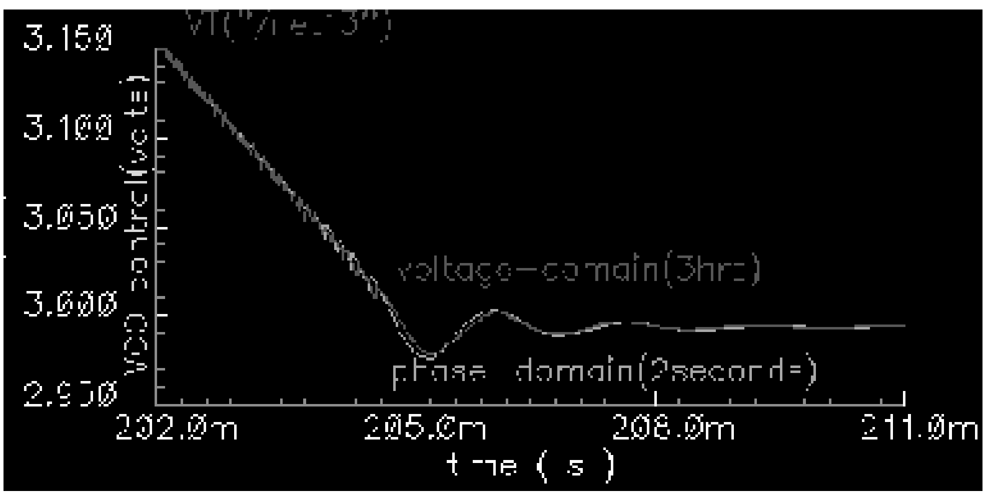
Use the *example_voltage_domain* model in the *pllLib* library with the only node that can be directly compared between phase- and voltage-domain models, the VCO control node. At full scale, the difference between the two models is not visible. The differences occur at the transitions. Figure 3-18 on page 434 compares the two models at the transitions. In this example and on the same machine, the voltage-domain model simulates in three hours while the phase-domain model simulates in two seconds.

The error between the two models does not appear to be consistent. It is larger in the first transition. Furthermore, decreasing the *speed_vs_accuracy* parameter does not always increase the similarity of the waveforms. This is because the final transient, the one driven primarily by phase error, depends on the residual frequency error at the time the phase error last crossed 2π. The frequency error at that moment, especially after a long frequency slewing period, is sensitive to, among other effects, initial conditions.

© 1994-2020
All Rights Reserved.

Figure 3-18 Comparison with Voltage Domain Model

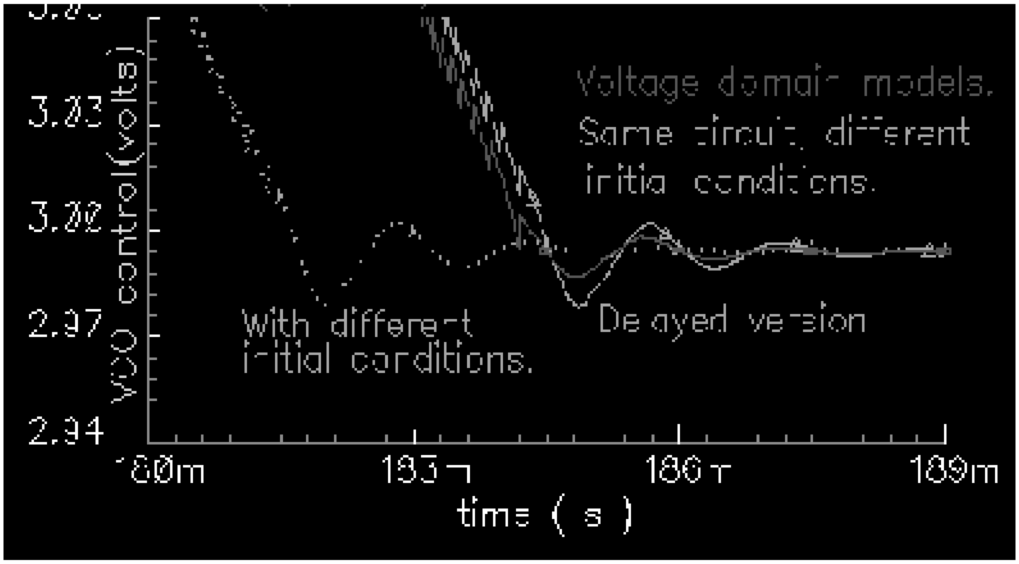




The voltage-domain model therefore shows the same level of variation for small differences in the initial conditions preceding the transition to the new equilibrium. Figure 3-19 on page 435 compares two voltage-domain simulations of the VCO control signal during the second transition. One of the voltage-domain simulations (the dotted waveform) used different initial conditions. The solid waveform is a delayed version of the dotted waveform. The delay overlays the two simulations for direct comparison. The error is comparable to the error between voltage-domain and phase-domain simulations.

© 1994-2020 All Rights Reserved.

Figure 3-19 Effect of Initial Conditions



How the PFD Model Works

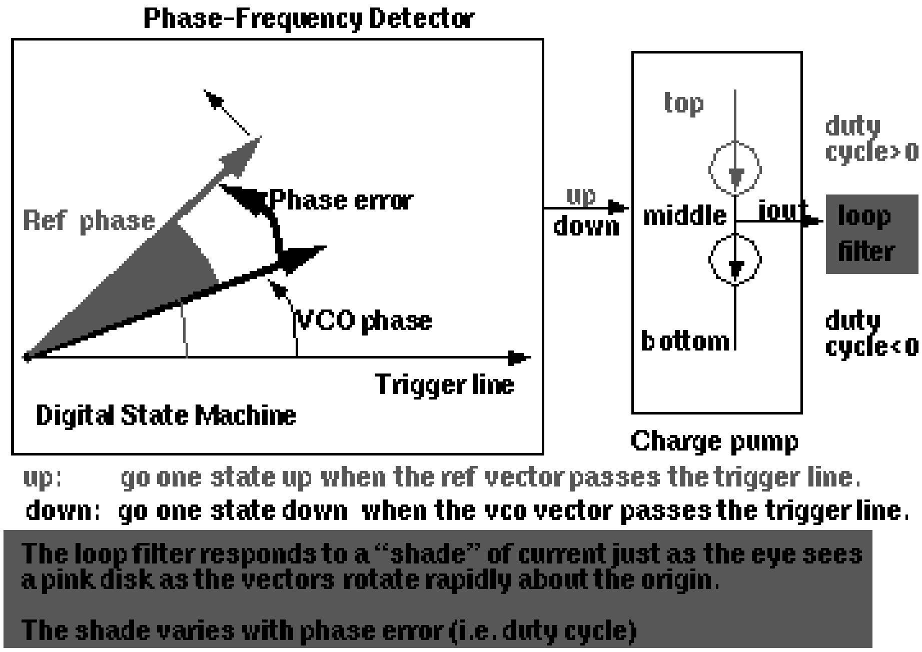
The heart of the PFD is a digital state machine [8,9].The model is for PFDs with three digital states. The output stage is usually a charge pump or pair of switches. It is convenient to model the PFD in two pieces. The first piece models the state machine and computes a duty cycle that is independent of the output stage. The second piece models the output stage. The charge pump (CP) is used here as an example.

How the PDF/CP Pump Works

Let the three PFD states be stacked. In the top state, the PFD commands the CP to source current. In the middle state, the pump is off. In the bottom state, the CP sinks current. The PFD is edge triggered. Figure 3-20 on page 436 shows vectorial representations of the reference and VCO clocks [1,4]. Both vectors rotate counter-clockwise around the origin. The angle between the hands equals phase error. Phase error lies between +-2π. Whenever the reference passes a trigger line, like 3 o'clock, the state jumps to the next state up. If the PFD is already in the top state, the state does not change. Whenever the VCO passes the trigger line, the state jumps to the next state down. If it is already in the bottom state, it again does not change. For a fixed phase error, the state toggles as the hands rotate. State toggles between the middle and top, or between the middle and bottom states. The percentage of time spent in the top, or bottom, state is the duty cycle. The duty cycle is positive for top to middle toggling and negative for middle to bottom toggling.

© 1994-2020
All Rights Reserved.

Figure 3-20 PFD Operation

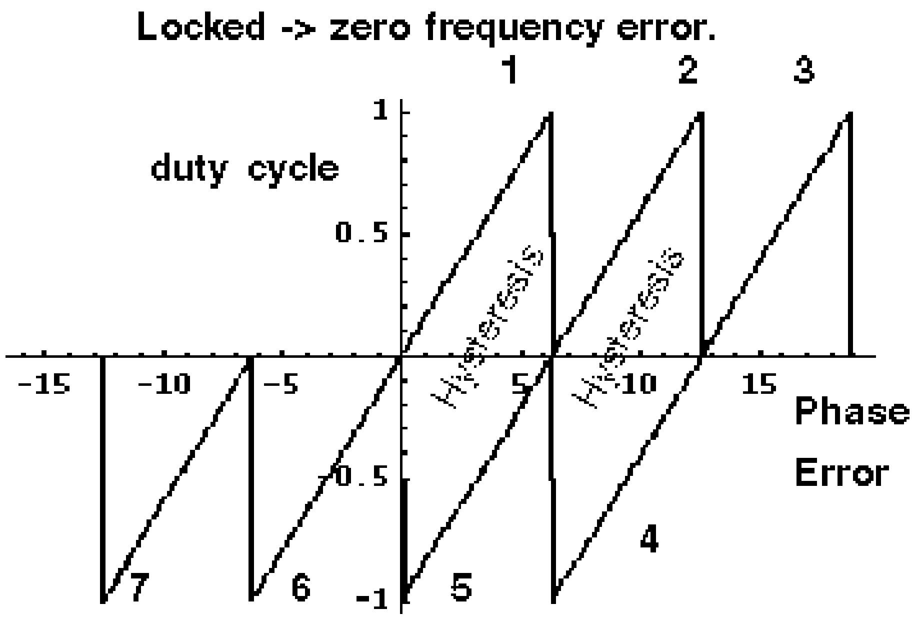


If the reference and VCO frequencies are identical, the vectors in Figure 3-20 rotate together. Let the sector defined by phase error be red if the reference leads and blue if it lags. If the hands rotate once per minute and the reference leads, you see two colors, white and red. At two million revolutions per second, you see pink. The shade of pink depends linearly on the phase error. Although PFD output current toggles between two values, the loop filter and VCO respond mainly to the "shade" of current. The shade is proportional to the duty cycle. With zero frequency error, duty cycle equals phase error divided by 2π. Existing literature uses one function to describe the *phase-error-to-duty-cycle* relationship and a different function to describe the *frequency-error-to-duty-cycle* relationship. These two functions are the *locked duty cycle function* and *averaged unlocked duty cycle function*, respectively. The new model combines these two functions into one practical model.

The locked duty cycle function is a multivalued sawtooth. For monotonic movements away from the origin of steady-state phase error, the duty cycle is a sawtooth in the upper-half plane. The duty cycle lies in the lower-half plane for negative movements. If a movement starts off positive, then changes direction, the duty cycle crosses zero and becomes a sawtooth in the lower half plane. The duty-cycle-phase-error trajectory encloses a nonzero area as shown by the {1-2-3-4-5-6-7} sequence of peaks in Figure 3-21 on page 437. This is a good reason for putting the integrator next to the non-linearity—hysteresis involves memory and the integral supplies it.

© 1994-2020 All Rights Reserved.

Figure 3-21 Duty Cycle Versus Phase Error With Zero Frequency Error



This can be modeled by a resettable integrator.

$$\textit{duty cycle} = \int_R \frac{(FrequencyError)}{2\pi}\,dt$$

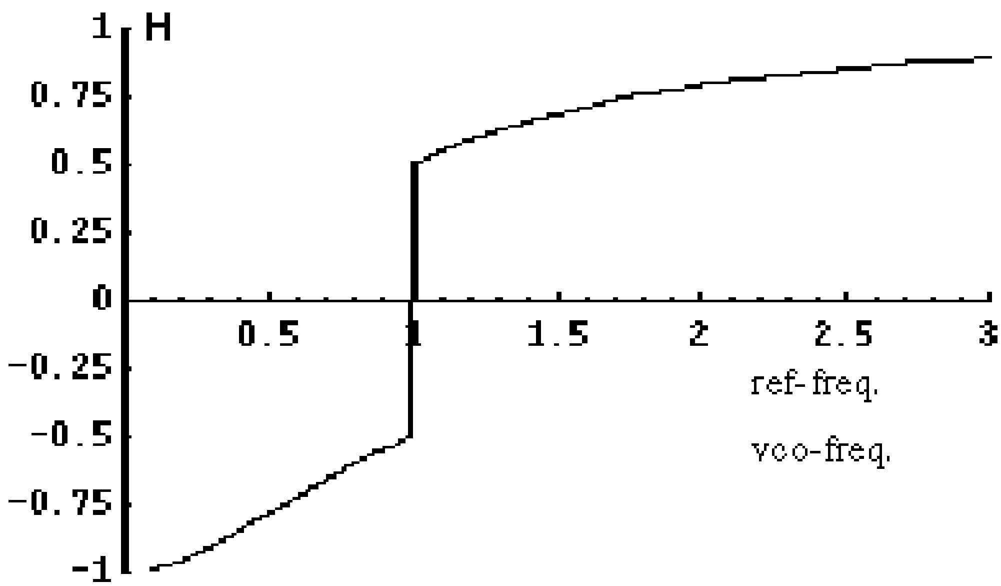
The new model operates only on frequency error. For small-frequency errors, the duty cycle is indeed proportional to the phase error. The phase error is the integral, with respect to time, of the frequency error. The duty cycle therefore equals the integral of the frequency error divided by 2π. Resetting the integral whenever it hits +-2π produces the multivalued sawtooth described above. If the frequency error changes sign, the resetting integrator ramps to zero, passes through zero, and generates a sawtooth in the lower-half plane. The phase-error-duty-cycle trajectory is precisely the multivalued sawtooth described above. The resettable integrator (RI) merges the integrator of a phase-domain model with the locked duty-cycle function.

For a sustained frequency error, the RI model predicts an average duty cycle of +-1/2 regardless of error size. This is correct only for small-frequency errors. The true duty cycle goes to +-1 for large-frequency errors. Let the reference frequency far exceed VCO frequency. Whenever the VCO passes the trigger line, the reference frequency passes shortly thereafter. The reference frequency might pass the trigger line several more times before the VCO passes again. In this case, the phase error is still a sawtooth, but the average duty cycle is nearly 1. This behavior lets the PLL acquire input signals faster. The function H, in Figure 3-

© 1994-2020 All Rights Reserved.

22 on page 438, shows the averaged unlocked duty cycle. This cycle depends on the ratio of the reference to VCO frequencies and is discontinuous where frequency error is zero.

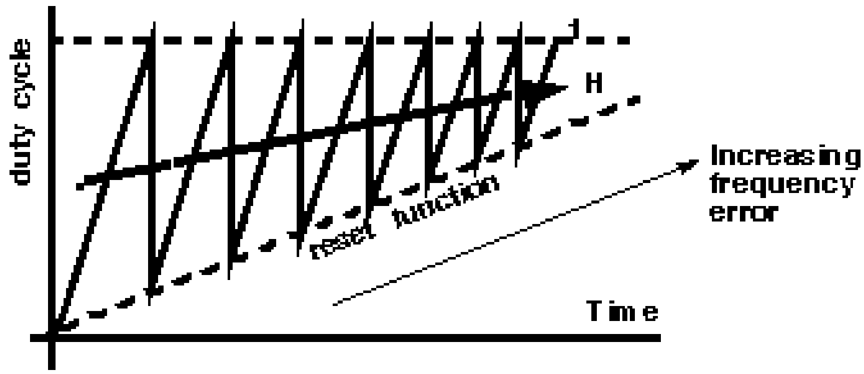
Figure 3-22 Averaged Unlocked Duty Cycle



The RI is modified to include the frequency effect. For small-frequency errors, the predicted average duty cycle equals 1/2. This is true because the RI runs from the reset point (=0) to the reset threshold (=2π). It is not necessary to reset the integrator to zero. Resetting the integrator to a “reset” function gives the correct average duty cycle (Figure 3-23). As the frequency ratio goes to +-infinity, the reset point changes to +-2π. Because the reset threshold is still +-2π, the predicted average duty cycle changes to +-1. As the frequency ratio approaches unity, the reset point returns to zero, and the predicted average duty cycle returns to 1/2.

© 1994-2020
All Rights Reserved.

Figure 3-23 Combining Averaged Locked and Unlocked Duty Cycles

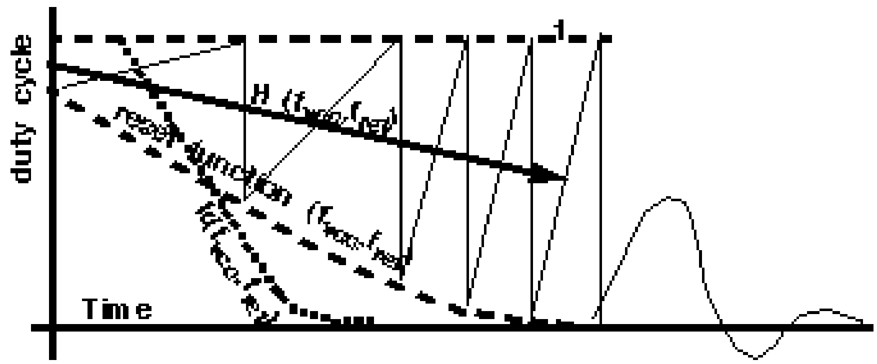


$$\text{duty cycle} = \int_{R(frequncyerror)} \frac{(FrequencyError)}{2\pi} dt$$

The state space averaged PFD model requires one more addition to be practical. As the reset point nears +-2π, the integrator resets very frequently and execution stalls. The integrator must be deactivated for large-frequency errors. The new PFD model uses the weighted sum of a RI and the H. The weighting factors are *k* and (1-*k*). *k* is a function of the ratio of the two input frequencies. *k* approaches 1 for large-frequency errors and approaches 0 as the frequency ratio approaches unity. A factor of (1-*k*) under the integral deactivates the integral for large-frequency errors. The resulting PFD model looks like H for large-frequency errors, and it looks like the reactivated RI for small-frequency errors. *k* determines how fast the RI reactivates and how gradually the model changes from H to the RI. A *speed_vs_accuracy* parameter controls *k*. If the model does not reset a few times before reaching frequency lock, you can improve the results by decreasing the *speed_vs_accuracy* parameter. If the model resets so often that the simulation is too slow, you can speed execution by increasing the parameter. The default setting of 50000 covers a wide range of loop speeds. Figure 3-24 on page 440 shows the transition from frequency-mode to phase-mode.

© 1994-2020
All Rights Reserved.

Figure 3-24 Complete Model



$$duty\ cycle = k*H + (1-k)\int_R (1-k)\frac{(FrequencyError)}{2\pi}dt$$

References

[1“Loops, Theory and Application,“ J. L. Stensby.

[2]“Relative -phase modeling speeds Spice simulation of modulated systems,” John Kesterson, November 11, 1993 issue of EDN

[3]“Simulation of Communication Systems,” M. Jeruchim, P. Balaban and K. Shanmugan. Plenum, 1992.

[4]“Synchronization in Digital Communications,” Heinrich Meyr and Gerd Ascheid, Published by John Wiley and Sons.

[5]“Phase-Locked and Frequency-Feedback Systems,” Klapper and Frankle, Published by Academic Press

[6]“Phaselock Techniques,” Floyd Gardner, Published by John Wiley and Sons

[7]“Non-Linear Relative Phase Models of Phaselock Loops,” Jess Chen, Cadence Technical Conference, 1996.

[8]“Phase-Locked Loops: Theory, Design, and Applications,” Roland Best, Published by McGraw-Hill

[9]“Phase-Locked Loop Circuit Design,” Dan Wolaver, Published by Prentice Hall

© 1994-2020 All Rights Reserved.

[10]“Phase-Lock Basics,” William F. Egan. Published by John Wiley and Sons.

[11]“Non-Linear State Space Averaged Modeling of a 3-State Digital Phase-Frequency Detector,” Jess Chen, Cadence Technical Conference, 1997

[12] “Analog and Mixed-Signal Hardware Description Languages,” Edited by A. Vachoux, J. Berge, O. Levia, and J. Rouillard. Kluwer Academic Publishers.

[13] “Macromodeling with SPICE,” J.A. Connelly and P. Choi. Prentice Hall. Pages 168-169.

[14] S. Cuk, California Institute of Technology, Ph.D. Thesis, “Modelling, Analysis, and Design of Switching Converters.” 1977

© 1994-2020 All Rights Reserved.