



Agilent Technologies

System Models

May 2003

Notice

The information contained in this document is subject to change without notice.

Agilent Technologies makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Agilent Technologies shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Warranty

A copy of the specific warranty terms that apply to this software product is available upon request from your Agilent Technologies representative.

Restricted Rights Legend

Use, duplication or disclosure by the U. S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 for DoD agencies, and subparagraphs (c) (1) and (c) (2) of the Commercial Computer Software Restricted Rights clause at FAR 52.227-19 for other agencies.

Agilent Technologies
395 Page Mill Road
Palo Alto, CA 94304 U.S.A.

Copyright © 1998-2003, Agilent Technologies. All Rights Reserved.

Acknowledgments

Mentor Graphics is a trademark of Mentor Graphics Corporation in the U.S. and other countries.

Microsoft®, Windows®, MS Windows®, Windows NT®, and MS-DOS® are U.S. registered trademarks of Microsoft Corporation.

Pentium® is a U.S. registered trademark of Intel Corporation.

PostScript® and Acrobat® are trademarks of Adobe Systems Incorporated.

UNIX® is a registered trademark of the Open Group.

Contents

1 Amplifiers and Mixers

Introduction.....	1-1
Curve-Fitting Algorithm.....	1-1
Amplifier (RF System Amplifier)	1-3
AmplifierVC (Ideal Voltage-Controlled Amplifier).....	1-11
AmpSingleCarrier (Single Carrier Amplifier).....	1-12
FreqMult (Ideal Frequency Multiplier)	1-13
LogACDemod (Demodulating AC Logarithmic Amplifier)	1-15
LogDC (DC Logarithmic Amplifier)	1-16
LogSuccDetect (Successive Detection Logarithmic Amplifier)	1-17
LogTrue (True Logarithmic Amplifier).....	1-18
Mixer (First RFSYSTEM Mixer, Polynomial Model for Nonlinearity).....	1-19
Mixer2 (Second RFSYSTEM Mixer, Polynomial Model for Nonlinearity).....	1-23
OpAmp (Operational Amplifier)	1-27
OpAmpIdeal (Ideal Operational Amplifier).....	1-30
VMult (Voltage Multiplier).....	1-32

2 Filters

Introduction.....	2-1
Filter Categories	2-2
BPF_Bessel (Bandpass Filter, Bessel-Thompson).....	2-5
BPF_Butterworth (Bandpass Filter, Butterworth)	2-6
BPF_Chebyshev (Bandpass Filter, Chebyshev).....	2-7
BPF_Elliptic (Bandpass Filter, Elliptic).....	2-9
BPF_Gaussian (Bandpass Filter, Gaussian)	2-10
BPF_PoleZero (Bandpass Filter, Pole Zero).....	2-11
BPF_Polynomial (Bandpass Filter, Polynomial).....	2-13
BPF_RaisedCos (Bandpass Filter, Raised-Cosine)	2-14
BSF_Bessel (Bandstop Filter, Bessel-Thompson)	2-17
BSF_Butterworth (Bandstop Filter, Butterworth)	2-18
BSF_Chebyshev (Bandstop Filter, Chebyshev).....	2-19
BSF_Elliptic (Bandstop Filter, Elliptic)	2-20
BSF_Gaussian (Bandstop Filter, Gaussian).....	2-21
BSF_PoleZero (Bandstop Filter, Pole Zero)	2-22
BSF_Polynomial (Bandstop Filter, Polynomial)	2-23
BSF_RaisedCos (Bandstop Filter, Raised-Cosine)	2-25
HPF_Bessel (Highpass Filter, Bessel-Thompson)	2-28
HPF_Butterworth (Highpass Filter, Butterworth)	2-29
HPF_Chebyshev (Highpass Filter, Chebyshev).....	2-30
HPF_Elliptic (Highpass Filter, Elliptic)	2-31

HPF_Gaussian (Highpass Filter, Gaussian).....	2-32
HPF_PoleZero (Highpass Filter, Pole Zero)	2-33
HPF_Polynomial (Highpass Filter, Polynomial)	2-34
HPF_RaisedCos (Highpass Filter, Raised-Cosine)	2-35
LPF_Bessel (Lowpass Filter, Bessel-Thompson).....	2-38
LPF_Butterworth (Lowpass Filter, Butterworth).....	2-39
LPF_Chebyshev (Lowpass Filter, Chebyshev)	2-40
LPF_Elliptic (Lowpass Filter, Elliptic).....	2-42
LPF_Gaussian (Lowpass Filter, Gaussian)	2-43
LPF_GMSK (Lowpass Filter, GMSK)	2-44
LPF_PoleZero (Lowpass Filter, Pole Zero).....	2-46
LPF_Polynomial (Lowpass Filter, Polynomial).....	2-47
LPF_RaisedCos (Lowpass Filter, Raised-Cosine)	2-48
SAW_Filter (Saw Filter)	2-51

3 Modulators and Demodulators

Introduction.....	3-1
AM_DemodTuned (AM Demodulator, Tuned).....	3-3
AM_ModTuned (AM Modulator, Tuned)	3-4
FM_DemodTuned (FM Demodulator, Tuned)	3-5
FM_ModTuned (FM Modulator, Tuned)	3-7
IQ_DemodTuned (I/Q Demodulator, Tuned)	3-9
IQ_ModTuned (I/Q Modulator, Tuned)	3-10
N_StateDemod (N-State Demodulator).....	3-11
N_StateMod (N-State Modulator)	3-12
PI4DQPSK_ModTuned (PI/4 DQPSK Modulator, Tuned).....	3-15
PM_DemodTuned (PM Demodulator, Tuned).....	3-17
PM_ModTuned (PM Modulator, Tuned)	3-18
PM_UnwrapDemodTuned (PM Unwrapped Demodulator, Tuned)	3-19
QPSK_ModTuned (QPSK Modulator, Tuned).....	3-20

4 Passive System Components

AntLoad (Antenna Load)	4-2
Attenuator (Attenuator)	4-4
Balun3Port (Balun, 3-port).....	4-6
Balun4Port (Balun, 4-port).....	4-8
Balun6Port (Balun, 6-port).....	4-10
Circulator (Ideal 3-Port Circulator).....	4-11
CouplerDual (Dual Coupler)	4-13
CouplerSingle (Single Coupler)	4-15
Gyrator (Gyrator)	4-17
Hybrid90 (Ideal 90-degree Hybrid Coupler).....	4-18
Hybrid180 (Ideal 180-degree Hybrid Coupler).....	4-20

IsolatorSML (SMLIsolator)	4-22
LOS_Link (Line-Of-Sight Antenna Link)	4-24
Pad (Pi or Tee Format)	4-26
PhaseShiftSML (Phase Shifter)	4-28
PwrSplit2 (2-Way Power Splitter)	4-30
PwrSplit3 (3-Way Power Splitter)	4-31
TimeDelay (Time Delay)	4-32
Transformer (Ideal 4-Port Transformer)	4-33
TransformerG (Transformer with Ground Reference)	4-34
TwoPort (2-Port Model)	4-35
5 Phase Lock Loop Components	
DivideByN (Divide by N)	5-2
PhaseFreqDet (Frequency Detector, Baseband)	5-5
PhaseFreqDet2 (Frequency Detector, Baseband)	5-9
PhaseFreqDetCP (Frequency Detector, Baseband with Charge Pump)	5-13
PhaseFreqDetTuned (Phase Frequency Detector, Tuned)	5-14
PhaseNoiseMod (Phase Noise Modulator)	5-17
VCO (Voltage Controlled Oscillator)	5-19
VCO_DivideByN (VCO Divide By N)	5-22
6 Switch and Algorithmic Components	
Comparator (Comparator)	6-2
ClockLFSR (Linear Feedback Shift Register)	6-3
Differentiator (Differentiator)	6-8
DPDT_Static (Double Pole Double Throw Switch, Static)	6-9
IntegratorSML (Integrator)	6-11
LimiterSML (Limiter)	6-12
ParallelSerial (Parallel to Serial Shift Register)	6-14
QuantizerSML (Quantizer)	6-15
ResetSwitch (Reset Switch)	6-17
SampleHoldSML (Sample Hold)	6-18
Sampler (Sampler)	6-19
SerialParallel (Serial to Parallel Shift Register)	6-24
SPDT_Dynamic (Single Pole Double Throw Switch, Dynamic)	6-25
SPDT_Static (Single Pole Double Throw Switch, Static)	6-27
SwitchV (Voltage Controlled Switch)	6-29
SwitchV_Model (Voltage Controlled Switch Model)	6-32
VSum (Voltage Summer)	6-33
7 Tx/Rx Subsystems	
RF_PA_CKT (RF Power Amplifier Circuit)	7-2
RF_RX_SML (RF Receiver)	7-3

RF_TX_SML (RF Transmitter)	7-4
----------------------------------	-----

8 System Data Models

Introduction.....	8-1
Amph1H2 (Amplifier/Fundamental and 2nd Harmonic vs. Input Power).....	8-3
Amph1H2_Setup (Amplifier/Fundamental and 2nd Harmonic vs. Input Power Setup).....	8-4
AmplifierP2D (P2D File Amplifier; FDD-Based, for Single Carrier Signal)	8-5
AmplifierP2D_Setup (P2D File Amplifier; FDD-Based, for Single Carrier Signal Setup).....	8-7
AmplifierS2D (S2D File Amplifier, Polynomial Model for Nonlinearity)	8-9
AmplLoadPull (SDD Load-Pull Amplifier)	8-11
IQ_Demod_Data (IQ Demodulator Behavioral Model)	8-12
IQ_Demod_Setup (IQ Demodulator Setup).....	8-14
IQ_Mod_Data (IQ Modulator Behavioral Model)	8-16
IQ_Mod_Setup (IQ Modulator Setup).....	8-18
LoadPullSetup (Load Pull Setup)	8-20
MixerHBdata (2-Tone HB Mixer).....	8-22
MixerHBsetup (2-Tone HB Mixer Setup).....	8-25
MixerIMT (Obsolete Intermodulation Table Mixer).....	8-26
MixerIMT2 (Intermodulation Table Mixer)	8-32
Intermodulation Product Table File Used with MixerIMT and MixerIMT2	8-35
VCA_Data (Voltage Controlled Amplifier)	8-38
VCA_Setup (Voltage Controlled Amplifier Setup).....	8-41

Index

Chapter 1: Amplifiers and Mixers

Introduction

The *Filters* - <filter type> and *System* - <device type> palettes contain two fundamentally different types of behavioral system models.

Filters, *System - Amps & Mixers*, and *System - Mod/Demod* can be classified as *tops-down system models* that support a tops-down system design flow where model behaviors are characterized by a small number of independent parameters such as frequency, power and load. They are often referred to as *parameter-based behavioral models*.

System - Data Models can be classified as *bottoms-up system models* that support a bottoms-up verification flow where model behaviors are extracted from a simulation (or measurement) of a transistor-level circuit. They are often referred to as *data-based behavioral models*.

The *parameter-based behavioral models* typically provide superior speed relative to the data-based behavioral models with both of these being vastly superior to a brute-force transistor-level simulation.

The *data-based behavioral models* typically provide superior accuracy relative to the *parameter-based behavioral models* as they capture actual behaviors of implemented circuit components and not just design specifications.

The differences between *parameter-* and *data-based behavioral models* justify a palette emphasis on flow (all data-based behavioral models grouped together) rather than functionality (all amplifiers, mixers, modulators, and demodulators grouped together) and resulted in the addition of a *System - Data Models* palette.

The use model for parameter-based behavioral models is to simply set a series of parameters prior to using the model. The use model for data-based behavioral models is slightly more involved. For a discussion, refer to Chapter 8, System Data Models.

Curve-Fitting Algorithm

The curve-fitting algorithm to determine the nonlinear behavior of the system mixer and amplifier models is based on fitting a polynomial to the specified data where the saturation power (P_{sat}) is calculated when the derivative of this polynomial is zero.

$$P_n(x) = a_1*x^1 + a_2*x^2 + a_3*x^3 + \dots$$

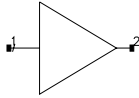
It is important to note that the coefficients a_4, a_6, a_8, \dots are always zero. In only one case a_2 is non-zero and that's when SOI and TOI are specified.

Here are the different scenarios and what order of polynomial used to curve-fit.

Parameters	Order
TOI	3
TOI & AM2PM	3
SOI & TOI	3 $a_2 \neq 0$
PndB	3
PndB & AM2PM	3
Psat	5
TOI & PndB	5
PndB & Psat	7
TOI & Psat	7
PndB & TOI & Psat	9

Amplifier (RF System Amplifier)

Symbol



Parameters

S21 = forward transmission gain (real or complex number; refer to note 2)

S11 = port 1 reflection (real or complex number; refer to note 2)

S22 = port 2 reflection (real or complex number; refer to note 2)

S12 = reverse transmission gain (real or complex number; refer to note 2)

NF = input noise figure, in dB

NFmin = minimum noise figure at Sopt, in dB

Sopt = optimum source reflection for NFmin

Rn = equivalent noise resistance

Z1 = reference impedance for port 1 (real or complex number)

Z2 = reference impedance for port 2 (real or complex number)

ClipDataFile = clip data beyond maximum input power: NO (default) to disable; YES to enable

ImpNoncausalLength = non-causal function impulse response order

ImpMode =convolution mode

ImpMaxFreq = maximum frequency to which device is evaluated

ImpDelta Freq = sample spacing in frequency

ImpMaxOrder = maximum allowed impulse response order

ImpWindow = smoothing window

ImpRelTol = relative impulse response truncation factor

ImpAbsTol = absolute impulse response truncation factor

Range of Usage

$NF \geq 0$ dB

$NF_{min} > 0$

$0 < |S_{opt}| < 1$

$0 < R_n$

$GainCompFreq > 0$

For $S_{21} = mag/ang$

$|S_{21}| > 0$

Gain Compression Parameters

GainCompType = gain compression type:

LIST, use model gain compression specifications

FILE, use file-based gain compression data

GainCompFreq = reference frequency for gain compression (if gain compression is described as a function of frequency)

ReferToInput = specify gain compression with respect to input or output power of device

SOI = second order intercept, in dBm

TOI = third order intercept, in dBm

Psat = power level at saturation, in dBm

GainCompSat = gain compression at Psat, in dB

GainCompPower = power level in dBm at gain compression for X dB compression point, specified by GainComp, in dBm

GainComp = gain compression at GainCompPower, in dB (default is 1dB)

AM2PM = amplitude modulation to phase modulation, in degree/dB

PAM2PM = power level at AM2PM, in dBm

GainCompFile = filename for gain compression data in S2D file format

Range of Usage for Gain Compression Parameters

When specifying gain compression using model parameters, only certain combination of parameters will produce stable polynomial curve fitting. The recommended parameter combinations are listed here.

Note If unrealistic parameter values are used, the polynomial will become unstable, resulting in oscillations.

- Third-order intercept and 1dB gain compression parameters:
TOI, GainCompPower with GainComp=1dB
Range of validity: TOI > GainCompPower + 10.8
- Third-order intercept and power saturation parameters:
TOI, Psat, GainCompSat
Range of validity: TOI > Psat + 8.6
- 1dB gain compression and power saturation parameters:
GainCompPower with GainComp=1dB, Psat, GainCompSat
Range of validity: Psat > GainCompPower + 3
- Third-order intercept, 1dB gain compression and power saturation parameters:
TOI, GainCompPower with GainComp=1dB, Psat, GainCompSat
Range of validity: Psat > GainCompPower +3, TOI > GainCompPower + 10.8
- Second-order intercept and third-order intercept parameters: SOI, TOI
- AM to PM with 1dB gain compression parameters:
AM2PM, PAM2PM, and GainCompPower with GainComp=1dB

The value for AM2PM must satisfy this condition to avoid a square root of a negative number:

$$AM2PM < \frac{180}{\pi} \times 10^{((PAM2PM - GainCompPower)/10)} \times 10^{(-3 \times (GainComp)/20)} \frac{1}{2.34}$$

- AM to PM with third-order intercept parameters:
AM2PM, PAM2PM, and TOI

The value for AM2PM must satisfy this condition to avoid a square root of a negative number:

$$AM2PM < \frac{\left(\frac{180}{\pi}\right) \times 10^{((PAM2PM - TOI)/10)}}{2.34}$$

If SOI is not specified, the amplifier is modeled using a polynomial of odd orders:

$$y = a1 \times x + a3 \times x^3 + a5 \times x^5 + \dots$$

As a result, only odd order harmonics ($m \times f$, where m is an odd number) and odd order intermods ($m \times f1 + n \times f2$, where $m+n$ is an odd number) are taken into account.

If SOI is specified, the amplifier polynomial has an even order term:

$$y = a1 \times x + a2 \times x^2 + a3 \times x^3 + a5 \times x^5 + \dots$$

As a result, both odd and even order harmonics and intermods are taken into account in the simulation.

Warning Messages

When values for TOI, 1 GainCompPower, and Psat are properly related, the DC input-output transfer characteristic has the form shown in [Figure 1-1](#).

- **No Saturation.** A warning is displayed if a polynomial is generated that does not have a maximum where the transfer characteristics can be clipped (when the amplifier cannot reach saturation). Refer to [Figure 1-2](#).
- **Non-Monotonic Transfer Curve.** A warning is displayed if the value specified for Psat is lower than GainCompPower. Refer to [Figure 1-3](#).

The result of this specification is that the saturated output is lower than the output at the 1 dB compression point, and the input-output characteristics have a non-monotonic characteristic transfer curve.

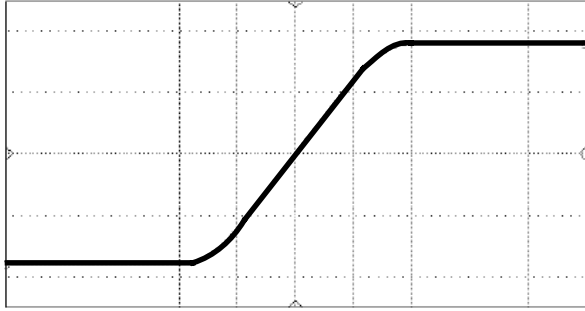


Figure 1-1. DC Input-Output Transfer Characteristics

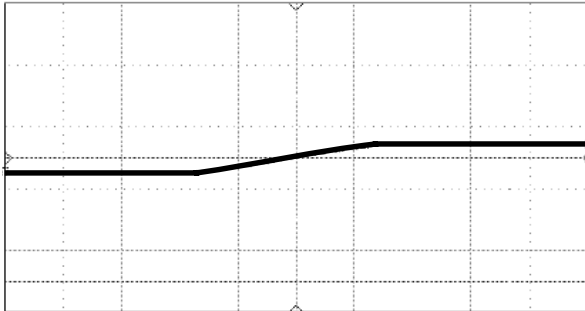


Figure 1-2. No Saturation Warning

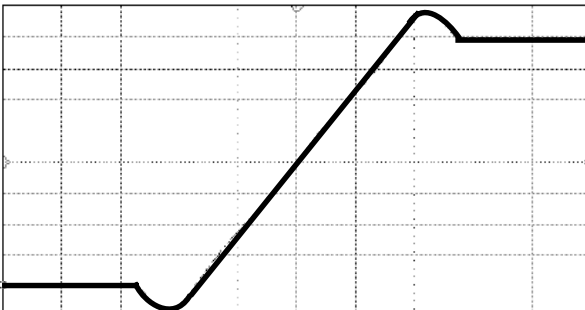


Figure 1-3. Non-Monotonic Transfer Curve Warning

Notes/Equations

1. If NFmin, Sopt, and Rn are used to characterize noise, the following relation must be satisfied for a realistic model.

$$\frac{Rn}{Zo} \geq \frac{To(Fmin-1)|1 + Sopt|^2}{T^4} \frac{(1 - |S_{11}|^2)}{|1 - Sopt S_{11}|^2}$$

A warning message will be issued if Rn does not meet this criterion. If the noise parameters attempt to describe a system that requires negative noise (due to Rn being too small), the negative part of the noise will be set to zero and a warning message will be issued.

2. Use the function polar(mag,ang), or dbpolar(dB, ang), or VSWRpolar(VSWR, ang) to convert these specifications into a complex number.
3. For an S-parameter or a noise figure sinusoidal ripple, use the function ripple(mag, intercept, period, variable); for example ripple(0.1, 0, 10 MHz, freq).
example: S21=dbpolar(10+ripple(),0.)
4. When you define the gain using S21, keep in mind that this gain is applied to the forward incident wave into the input of the amplifier. This is in keeping with the measurement standards used to define amplifier gain at a system level. This means that if you change S11 from 0 to 0.9 for example, you will see no change in output power because the reflect wave is not taken into account by the amplifier's definition of gain, only the incident wave.
5. This model blocks dc.
6. For circuit envelope simulation, baseband signals are blocked.
7. OmniSys used GComp1-GComp7 data items for specifying gain compression. [Table 1-1](#) summarizes the gain compression data for OmniSys and ADS. Refer to [Figure 1-4](#) for OmniSys parameter information.

GComp1-GComp6 can be specified by using the corresponding ADS gain compression parameters and setting GainCompType=LIST. Or, they can also be contained in an S2D format setting GainCompType=FILE. Also note that an S2D file could contain other data such as small signal S-parameters and noise; these data are ignored by the RF System Amplifier.

Table 1-1. Gain Compression Data for OmniSys and ADS

OmniSys	ADS
GComp1: IP3	TOI
GComp2: 1dBc	GainComp=1dB GainCompPower
GComp3: IP3, 1dBc	TOI GainComp=1dB GainCompPower
GComp4: IP3, Ps, GCS	TOI Psat GainCompSat
GComp5: 1dBc, Ps, GCS	GainComp=1dB GainCompPower Psat GainCompSat
GComp6: IP3, 1dBc, Ps, GCS	TOI GainComp=1dB GainCompPower Psat GainCompSat
GComp7	GainCompType=FILE GainCompFile=filename

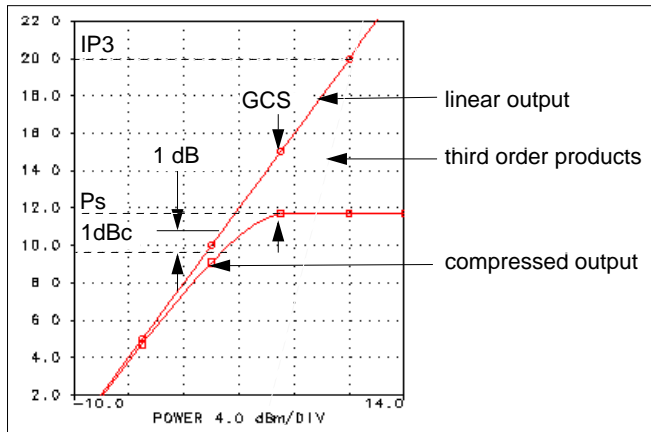


Figure 1-4. OmniSys Parameter Information

8. The AM to PM option uses parabolic amplitude dependence to describe the amplitude to phase modulation conversion. When a signal of type

$V_{in}(\tau) = A \cos(W\phi\tau)$, is applied to the input of a device with parabolic AM to PM, the output phase exhibits:

$$Phase(V_{out}(\tau)) = kA^2$$

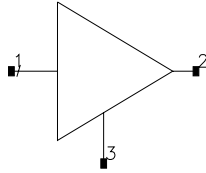
Therefore, this phase depends on the input signal amplitude A in a parabolic manner. Because the conversion of amplitude to phase is amplitude dependent, the AM to PM (AM2PM) is specified in degrees per decibel at a given output power (PAM2PM). k is calculated from these two parameters.

$$k = \frac{AM2PM \left(\frac{\pi}{180} \right)^{2.34}}{10^{\left(\frac{PAM2PM - 30}{10} \right)}}$$

When AM to PM is specified, the third-order intermod and gain compression are side effects. If AM2PM is specified to be very large compared to the third-order intercept or gain compression, a warning is issued.

AmplifierVC (Ideal Voltage-Controlled Amplifier)

Symbol



Parameters

Gain = voltage gain, in dB/V

Rout = output resistance, in ohms

Range of Usage

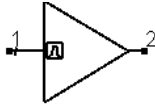
Rout > 0

Notes/Equations

1. AmplifierVC is an ideal voltage controlled amplifier. The impedance for its input and control port is infinite; its output impedance is Rout.
2. An equation is used to describe Gain as a function of the control voltage at port 3: $_V3$. The default equation is $\text{Gain} = 30 - 15 \times _V3$.

AmpSingleCarrier (Single Carrier Amplifier)

Symbol



Parameters

Freq = frequency

S21 = forward transmission coefficient; use $x+j*y$ polar (x,y), dbpolar (x, y) for complex value

S11 = forward reflection coefficient; use $x+j*y$ polar (x,y), dbpolar (x, y), vswrpolar (x, y) for complex value

S22 = reverse reflection coefficient; use $x+j*y$ polar (x,y), dbpolar (x, y), vswrpolar (x, y) for complex value

S12 = reverse reflection coefficient; use $x+j*y$ polar (x,y), dbpolar (x, y) for complex value

NF = noise figure, in dB

NFmin = minimum noise figure at Sopt, in dB

Sopt = optimum source reflection for minimum noise figure; use $x+j*y$ polar (x,y), dbpolar (x, y) for complex value

Rn = equivalent noise resistance

Z1 = reference impedance for port 1

Z2 = reference impedance for port 2

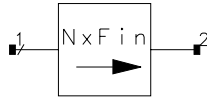
GainCompFile = filename for Gain Compression (GCOMP7) Data

Notes/Equations

1. AmpSingleCarrier is based on FDD (frequency-domain defined device). The incident power level at the amplifier input is detected and the corresponding gain is obtained by interpolating the gain compression table given by GainCompFile. In harmonic balance simulation, the output signal has only one frequency component given by the Freq parameter. Neither harmonics nor intermodulation products are generated by AmpSingleCarrier. In Envelope simulation, the output signal contains only the complex envelope around one frequency given by the Freq parameter.

FreqMult (Ideal Frequency Multiplier)

Symbol



Parameters

S11 = complex reflection coefficient for port 1

S22 = complex reflection coefficient for port 2

G1 = power gain of input tone, in dB

G2 = power gain of second harmonic relative to input tone, in dB

G3 = power gain of third harmonic relative to input tone, in dB

G4 = power gain of fourth harmonic relative to input tone, in dB

G5 = power gain of fifth harmonic relative to input tone, in dB

G6 = power gain of sixth harmonic relative to input tone, in dB

G7 = power gain of seventh harmonic relative to input tone, in dB

G8 = power gain of eighth harmonic relative to input tone, in dB

G9 = power gain of ninth harmonic relative to input tone, in dB

Pmin = minimum input power for specified conversion, in dBm

Z1 = reference impedance for port 1

Z2 = reference impedance for port 2

Range of Usage

$$0 \leq |S11| < 1$$

$$0 \leq |S22| < 1$$

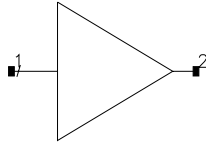
Notes/Equations

1. The ideal frequency multiplier takes an input signal and produces an output spectrum with specified spectral harmonics. The reverse isolation is assumed to be infinite ($S12=0$). All of the harmonics generation is specified relative to the input level. For example if an input power of 20 dBm is incident on a multiplier with $G2=-20$ dB the second harmonic output will be 0 dBm. This device is compatible with transient simulation.

2. This model assumes that only one signal tone is present at the input. If multiple tones are used at the input then unwanted mixing products can be generated and spurious mixing products will result.
3. The harmonic balance parameter ORDER must be set to a value equal to or higher than the harmonic index of interest.

LogACDemod (Demodulating AC Logarithmic Amplifier)

Symbol



Parameters

CurrentSlope = gradient of transfer characteristic, in amperes/decade

VoltIntercept = V_{in} for zero output, in volts

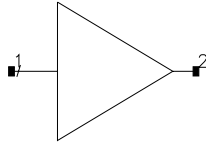
Z1 = reference impedance for port 1, in ohms

Notes/Equations

1. LogACDemod uses a square-law detector and an ideal logarithmic function.
2. The function is in the form of voltage-input/current-output.
3. LogACDemod is not recommended for transient simulation.

LogDC (DC Logarithmic Amplifier)

Symbol



Parameters

VoltSlope = gradient of transfer characteristic, in volts/decade

VoltIntercept = V_{in} for zero output, in volts

Z1 = reference impedance for port 1, in ohms

Notes/Equations

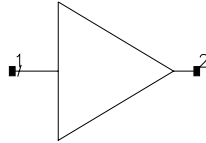
1. LogDC provides an output that is a logarithmic function of the input. If, for example, the scaling is 1 volt/decade, the output changes by 1V for any tenfold increase of the input.

The intercept point is the input level at which the output voltage is 0.

$$V_{out} = \text{VoltSlope} \times \log \left(\left(V_{in} / \text{VoltIntercept} \right) + 1 \right)$$

LogSuccDetect (Successive Detection Logarithmic Amplifier)

Symbol



Parameters

NumStages = number of stages

StageGain = gain per stage, in dB

CurrentSlope = gradient of transfer characteristic, in amperes/decade

Z1 = reference impedance for port 1, in ohms

Notes/Equations

1. This amplifier uses a successive detection scheme to provide an output current proportional to the logarithm of the input voltage. The amplifier consists of several amplifier/limiter stages (NumStages specifies how many), each having a small signal gain (StageGain specifies the gain). Each stage has an associated full-wave detector, whose output current depends on the absolute value of its input stage. These output currents are summed to provide the output scaled at the CurrentSlope (amp/decade). The output contains amplitude information only, regardless of any phase information. A simplified block diagram of this component is shown in [Figure 1-5](#).

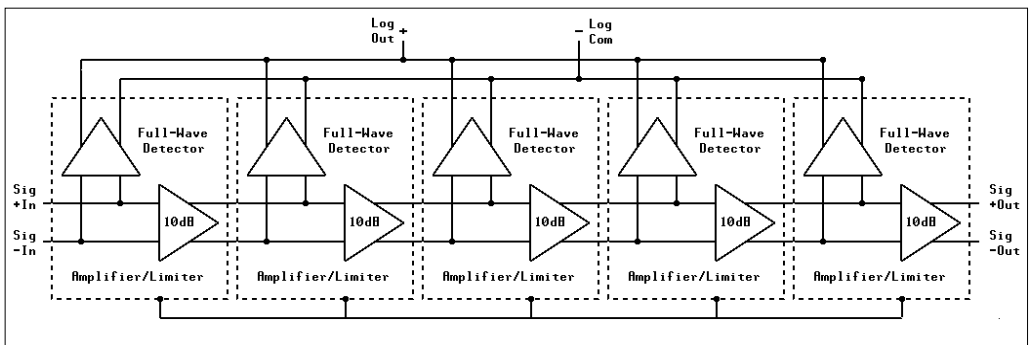
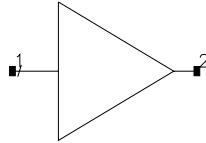


Figure 1-5. Simplified Block Diagram of a 5-Stage LogSuccDetect

LogTrue (True Logarithmic Amplifier)

Symbol



Parameters

NumStages = number of stages

StageGain = gain per stage, in dB

VoltLimit = limiting voltage of each stage, in volts

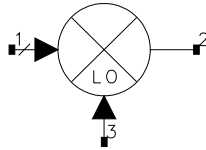
Z1 = reference impedance for port 1, in ohms

Notes/Equations

1. LogTrue accepts inputs of either polarity and generates an output whose sign follows that of the input. A progressive compression technique is used, in which the logarithmic response can be approximated through the use of cascaded amplifier stages that have signal-dependent gain.

Mixer (First RFSystem Mixer, Polynomial Model for Nonlinearity)

Symbol



Parameters

SideBand = produce UPPER, LOWER, or BOTH sidebands

ImageRej = image rejection at output with respect to fundamental, in dB

LO_Rej1 = LO to input rejection for LO leakage, in dB

LO_Rej2 = LO to output rejection for LO leakage, in dB

RF_Rej = input to output rejection for direct RF feedthrough, in dB

ConvGain = conversion gain (real or complex number; see note 2)

S11 = port 1 reflection (real or complex number; see note 2)

S22 = port 2 reflection (real or complex number; see note 2)

S33 = port 3 reflection (real or complex number; see note 2)

PminLO = minimum LO power before starvation, in dBm

NF = input double sideband noise figure, in dB

NFmin = minimum double sideband noise figure at Sopt, in dB

Sopt = optimum source reflection for NFmin

Rn = equivalent noise resistance, in ohms

Z1 = reference impedance for port 1 (real or complex number)

Z2 = reference impedance for port 2 (real or complex number)

Z3 = reference impedance for port 3 (real or complex number)

ImpNoncausal Length = non-causal function impulse response order

ImpMode = convolution mode

ImpMaxFreq = maximum frequency at which device is evaluated

ImpDeltaFreq = sample spacing in frequency

ImpMaxOrder = maximum allowed impulse response order

ImpWindow = smoothing window

ImpRelTol = relative impulse response truncation factor

ImpAbsTol = absolute impulse response truncation factor

Range of Usage

$NF \geq 0$ dB

$NF_{min} > 0$

$0 < |S_{opt}| < 1$

$0 < R_n$

GainCompFreq > 0

$|ConvGain| > 0$

Gain Compression Parameters

GainCompType = gain compression type: LIST, use model gain compression specifications; FILE, use file-based gain compression data

GainCompFreq = reference frequency for gain compression (if gain compression is described as a function of frequency)

ReferToInput = specify gain compression with respect to input or output power of device

SOI = second order intercept, in dBm

TOI = third order intercept, in dBm

Psat = power level at saturation, in dBm

GainCompSat = gain compression at Psat, in dB

GainCompPower = power level in dBm at gain compression specified by GainComp, in dBm

GainComp = gain compression at GainCompPower, in dB (default is 1dB)

GainCompFile = filename for gain compression data in S2D file format

Range of Usage for Gain Compression Parameters

When specifying gain compression using model parameters, only certain combination of parameters will produce stable polynomial curve fitting. Recommended parameter combinations are listed here.

Note If unrealistic parameter values are used, the polynomial will become unstable, resulting in oscillations.

- Third-order intercept and 1dB gain compression parameters: TOI, GainCompPower, with GainComp=1dB.
Range of validity: TOI > GainCompPower + 10.8.
- Third-order intercept and power saturation parameters: TOI, Psat, GainCompSat.
Range of validity: TOI > Psat + 8.6.
- 1dB gain compression and power saturation parameters: GainCompPower, with GainComp=1dB, and Psat, GainCompSat.
Range of validity: Psat > GainCompPower + 3.
- Third-order intercept, 1dB gain compression and power saturation parameters: TOI, GainCompPower with GainComp=1dB, and Psat, GainCompSat.
Range of validity: Psat > GainCompPower +3, TOI > GainCompPower + 10.8
- Second-order intercept and third-order intercept parameters: SOI, TOI.

Notes/Equations

1. The Mixer component is similar to Mixer2. The key difference is that Mixer supports frequency conversion AC analysis or FCAC analysis for small-signal AC or S-parameter analysis, while Mixer2 does not. This capability allows small-signal frequency traditionally done at only one frequency to be somewhat extended to deal with more than one frequency. In terms of convergence, Mixer2 is typically more robust than Mixer, as the power detection is implemented differently.
2. If NFmin, Sopt, and Rn are used to characterize noise, the following relation must be satisfied for a realistic model.

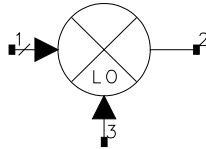
$$\frac{Rn}{Zo} \geq \frac{To(Fmin-1)|1 + Sopt|^2}{T^4} \frac{(1 - |S_{11}|^2)}{|1 - Sopt S_{11}|^2}$$

A warning message will be issued if Rn does not meet this criterion. If the noise parameters attempt to describe a system that requires negative noise (due to Rn being too small), the negative part of the noise will be set to zero and a warning message will be issued.

3. Use the function `polar(mag,ang)` or `dbpolar(dB, ang)`, or `VSWR polar(VSWR, ang)` to convert these specifications into a complex number.
4. For an S-parameter or a noise figure sinusoidal ripple, use the function `ripple(mag, intercept, period, variable)`; for example `ripple(0.1, 0, 10 MHz, freq)`.
Example: `S21=dbpolar(10+ripple(),0.)`
5. This model passes DC in the sense that a DC source at the RF input passes through the mixer to give a signal at the IF output.
6. Gain compression can be specified by using the gain compression model parameters, or this information can be contained in an S2D format file. All S2D gain compression types are supported by this model. Gain compression types 1 through 6 can also be described using the gain compression model parameters. Gain Compression 7 information must be contained in an S2D file. The `GainCompType` parameter instructs the model where to look for this data—in an S2D file or use model parameters.
For S2D data file format information refer to Chapter 7 *Working with Data Files* in the *Circuit Simulation* manual.
7. Frequency Conversion AC (FCAC) analysis requires a single frequency at every node of a circuit. If `Sideband=BOTH` is specified for Mixer, the simulator will use `SideBand=LOWER` for FCAC analysis.
8. When Mixer is used in Transient analysis for single sideband applications, it is recommended that the user set `Sideband=BOTH` and insert a high-order filter to suppress the undesired sideband.

Mixer2 (Second RFSystem Mixer, Polynomial Model for Nonlinearity)

Symbol



Parameters

SideBand = produce UPPER, LOWER, or BOTH sidebands

ConvGain = conversion gain (real or complex number; see note 2)

SP11 = RF port reflection; use $x+j*y$, polar (x,y), dbpolar (x,y), vswrpolar (x,y), for complex value

SP12 = IF port to RF port leakage; use $x+j*y$, polar (x,y), dbpolar (x,y), vswrpolar (x,y), for complex value

SP13 = LO port to RF port leakage; use $x+j*y$, polar (x,y), dbpolar (x,y), vswrpolar (x,y), for complex value

SP21 = RF port to IF port leakage; use $x+j*y$, polar (x,y), dbpolar (x,y), vswrpolar (x,y), for complex value

S22 = IF port reflection; use $x+j*y$, polar (x,y), dbpolar (x,y), vswrpolar (x,y), for complex value

SP23 = LO port to IF port leakage; use $x+j*y$, polar (x,y), dbpolar (x,y), vswrpolar (x,y), for complex value

SP31 = RF port to LO port leakage; use $x+j*y$, polar (x,y), dbpolar (x,y), vswrpolar (x,y), for complex value

SP32 = IF port to LO port leakage; use $x+j*y$, polar (x,y), dbpolar (x,y), vswrpolar (x,y), for complex value

SP33 = LO port reflection; use $x+j*y$, polar (x,y), dbpolar (x,y), vswrpolar (x,y), for complex value

PminLO = minimum LO power before starvation, in dBm

NF = input double sideband noise figure, in dB

NFmin = minimum double sideband noise figure at S_{opt} , in dB

Sopt = optimum source reflection for NFmin

Rn = equivalent noise resistance, in ohms

Z1 = reference impedance for port 1 (real or complex number)

Z2 = reference impedance for port 2 (real or complex number)

Z3 = reference impedance for port 3 (real or complex number)

ImpNoncausal Length = non-causal function impulse response order

ImpMode = convolution mode

ImpMaxFreq = maximum frequency at which device is evaluated

ImpDeltaFreq = sample spacing in frequency

ImpMaxOrder = maximum allowed impulse response order

ImpWindow = smoothing window

ImpRelTol = relative impulse response truncation factor

ImpAbsTol = absolute impulse response truncation factor

Range of Usage

$NF \geq 0$ dB

$NFmin > 0$

$0 < |Sopt| < 1$

$0 < Rn$

$GainCompFreq > 0$

$|ConvGain| > 0$

Gain Compression Parameters

GainCompType = gain compression type: LIST, use model gain compression specifications; FILE, use file-based gain compression data

ReferToInput = specify gain compression with respect to input or output power of device

SOI = second order intercept, in dBm

TOI = third order intercept, in dBm

Psat = power level at saturation, in dBm

GainCompSat = gain compression at Psat, in dB

GainCompPower = power level in dBm at gain compression specified by GainComp, in dBm

GainComp = gain compression at GainCompPower, in dB (default is 1dB)

AM2PM = amplitude to phase modulation in degrees, dB

PAM2PM = power level at AM2PM in degrees

GainCompFile = filename for gain compression data in S2D file format

Range of Usage for Gain Compression Parameters

When specifying gain compression using model parameters, only certain combination of parameters will produce stable polynomial curve fitting. The recommended parameter combinations are listed here.

Note If unrealistic parameter values are used, the polynomial will become unstable, resulting in oscillations.

- Third-order intercept and 1dB gain compression parameters:
TOI, GainCompPower, with GainComp=1dB.
Range of validity: $TOI > GainCompPower + 10.8$.
- Third-order intercept and power saturation parameters:
TOI, Psat, GainCompSat.
Range of validity: $TOI > Psat + 8.6$.
- 1dB gain compression and power saturation parameters: GainCompPower, with GainComp=1dB, and Psat, GainCompSat.
Range of validity: $Psat > GainCompPower + 3$.
- Third-order intercept, 1dB gain compression and power saturation parameters:
TOI, GainCompPower with GainComp=1dB, and Psat, GainCompSat.
Range of validity: $Psat > GainCompPower + 3$, $TOI > GainCompPower + 10.8$
- Second-order intercept and third-order intercept parameters: SOI, TOI.

Notes/Equations

1. The Mixer component is very similar to Mixer2. The key difference is that Mixer supports frequency conversion AC analysis or FCAC analysis for small-signal AC or S-parameter analysis, while Mixer2 does not. This capability allows small-signal frequency traditionally done at only one frequency to be

somewhat extended to deal with more than one frequency. In terms of convergence, Mixer2 is typically more robust than Mixer, as the power detection is implemented differently.

2. If NFmin, Sopt, and Rn are used to characterize noise, the following relation must be satisfied for a realistic model.

$$\frac{Rn}{Zo} \geq \frac{To(Fmin-1)|1 + Sopt|^2 (1 - |S_{11}|^2)}{T^4 |1 - Sopt S_{11}|^2}$$

A warning message will be issued if Rn does not meet this criterion. If the noise parameters attempt to describe a system that requires negative noise (due to Rn being too small), the negative part of the noise will be set to zero and a warning message will be issued.

3. Use the function polar(mag,ang) or dbpolar(dB, ang), or VSWR polar(VSWR, ang) to convert these specifications into a complex number.
4. For an S-parameter or a noise figure sinusoidal ripple, use the function ripple(mag, intercept, period, variable); for example ripple(0.1, 0, 10 MHz, freq).

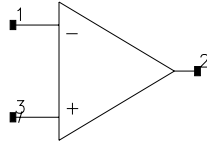
Example: S21=dbpolar(10+ripple(),0.)

5. This model passes DC in the sense that a DC source at the RF input is filtered out and does not pass through the mixer at the IF output. There can be a DC IF output signal for this mixer, but this results from the mixing signal and not a DC signal at FR input.
6. Gain compression can be specified by using the gain compression model parameters, or this information can be contained in an S2D format file. All S2D gain compression types are supported by this model. Gain compression types 1 through 6 can also be described using the gain compression model parameters. Gain Compression 7 information must be contained in an S2D file. The GainCompType parameter instructs the model where to look for this data—in an S2D file or use model parameters.

For S2D data file format information refer to Chapter 7 *Working with Data Files* in the *Circuit Simulation* manual.

OpAmp (Operational Amplifier)

Symbol



Parameters

Gain = open loop dc gain of amplifier, in dB

CMR = common mode rejection ratio, in dB

Rout = output resistance, in ohms

RDiff = differential input resistance, in ohms

CDiff = differential input capacitance, in farads

RCom = common mode input resistance, in ohms

CCom = common mode input capacitance, in farads

SlewRate = signal slew rate, in volts/sec

IOS = input offset current, in amperes

VOS = input offset voltage, in volts

BW = gain bandwidth product (unity gain bandwidth), in hertz

Pole1 = dominant pole frequency, in hertz (overrides BW parameter)

Pole2 = additional higher order pole frequency, in hertz

Pole3 = additional higher order pole frequency, in hertz

Pole4 = additional higher order pole frequency, in hertz

Pole5 = additional higher order pole frequency, in hertz

Zero1 = feed forward zero frequency, in hertz

Inoise = input spectral noise current, in amperes/sqrt(hertz)

Vnoise = input spectral noise voltage, in volts/sqrt(hertz)

VEE = negative supply voltage, in volts

VCC = positive supply voltage, in volts

DeltaVEE = difference between output saturation and VEE (see note 4)

DeltaVCC = difference between output saturation and VCC (see note 4)

Range of Usage

RDiff > 0

CDiff > 0

RCom > 0

CCom > 0

by default:

CMR = ∞

Pole2 = ∞

Pole3 = ∞

Pole4 = ∞

Pole5 = ∞

Zero1 = ∞

Notes/Equations

1. The BW parameter is the GainBandwidth product, i.e. it is the frequency at which the gain is unity or 0 dB. Pole1 is the basic amplifier pole and corresponds to the frequency where the gain starts sloping downward.
 BW and Pole1 can be specified simultaneously; however, if both are entered, Pole1 will override BW, and Pole1 must then be entered as BW/Gain. The Gain parameter is the open loop gain of the opamp and it must be converted out of dB only for use in setting Pole1, i.e., $\text{Gain} = 10^{\text{Gain_dB}/20}$.
2. To match the phase shift from the data sheet, adjust the values of Pole2 through Pole5.
3. Zero1 is used for operational amplifiers with feed-forward or lead-lag compensation networks.
4. Output voltage is generally less than the rail voltage (VCC and VEE). Use DeltaVCC and DeltaVEE to specify the difference between the rail voltage and actual output voltage. For example, if VCC is +5V and the positive output is +4.5V, set DeltaVCC to 0.5V.
5. This opamp is a nonlinear model. If your circuit cannot achieve convergence using this model, use the OpAmpIdeal linear model.
6. The relationship between input and output voltages is given in the equation:

$$I_m \times \tanh(V_{in}/I_m) = V_{out}/A_0 - 10 \times (V_{clip} - V_{out}) + d/dt (V_{out} \times \text{Tau1}/A_0)$$

where

A_0 is open loop DC gain

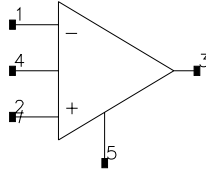
$V_{clip} = V_{out}$ as long as it is not limiting

$I_m = SlewRate \times \tau_1 / A_0$

$\tau_1 = A_0 / 2 / \pi / BW$ when $Pole_1 = 0$, otherwise $\tau_1 = 1 / 2 / \pi / BW$

OpAmpIdeal (Ideal Operational Amplifier)

Symbol



Parameters

Gain = magnitude of open loop dc voltage gain; use the polar() function to specify magnitude and phase

Z1 = input impedance, inverting terminal, in ohms

Z2 = input impedance, non-inverting terminal, in ohms

Z3 = output impedance, in ohms

Z4 = leakage impedance, inverting to non-inverting terminal, in ohms

Freq3db = frequency at which gain magnitude is down by 3dB, in hertz

Delay = time delay associated with gain, in seconds

Range of Usage

by default:

$$Z1 = \infty$$

$$Z2 = \infty$$

$$Z3 = 0$$

$$Z4 = \infty$$

$$\text{Freq3db} = \infty$$

Notes/Equations

$$1. \quad V_s = (V^+ - V^-) \text{Gain} \times \frac{e^{-j2\pi F \text{Delay}}}{1 + j \frac{F}{\text{Freq 3dB}}}$$

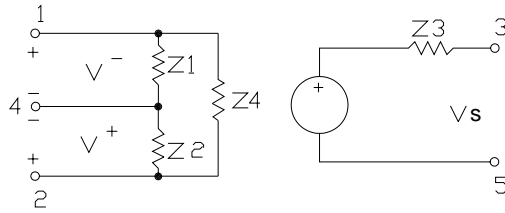
$$V_s = (V^+ - V^-) \text{Gain} \text{ (for } f=0\text{)}$$

where F is the simulation frequency

2. OpAmpIdeal is a noiseless component.

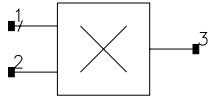
3. The recommendation is to use a Gain value no greater than $1e10$, or 200 dB for practical purposes.

Equivalent Circuit



VMult (Voltage Multiplier)

Symbol



Parameters

R1 = reference resistance for port 1, in ohms

R2 = reference resistance for port 2, in ohms

R3 = reference resistance for port 3, in ohms

L31 = loss in dB, pin 1 to pin 3; for linear analysis only

L32 = loss in dB, pin 2 to pin 3; for linear analysis only

Linear = yes enables linear analysis mode, for use with linear simulations

Notes/Equations

1. VMult uses reference input and output impedances. Its output voltage is equal to the product of its two input voltages.

2. If Linear is set to yes,

$$V_3 = \frac{A_{31} V_1 + A_{32} V_2}{R_3}$$

A31 and A32 are the L31 and L32 losses converted from dB to regular values.

If Linear is set to no, V3 is formed from V1 and V2 through an ideal mixing process; for information on this process refer to the Ideal Mixer section in Chapter 5 of the User-Defined Models manual.

Chapter 2: Filters

Introduction

The *Filters* - *<filter type>* and *System* - *<device type>* palettes contain two fundamentally different types of behavioral system models.

Filters, *System - Amps & Mixers*, and *System - Mod/Demod* can be classified as *tops-down system models* that support a tops-down system design flow where model behaviors are characterized by a small number of independent parameters such as frequency, power and load. They are often referred to as *parameter-based behavioral models*.

System - Data Models can be classified as *bottoms-up system models* that support a bottoms-up verification flow where model behaviors are extracted from a simulation (or measurement) of a transistor-level circuit. They are often referred to as *data-based behavioral models*.

The *parameter-based behavioral models* typically provide superior speed relative to the data-based behavioral models with both of these being vastly superior to a brute-force transistor-level simulation.

The *data-based behavioral models* typically provide superior accuracy relative to the *parameter-based behavioral models* as they capture actual behaviors of implemented circuit components and not just design specifications.

The differences between *parameter-* and *data-based behavioral models* justify a palette emphasis on flow (all data-based behavioral models grouped together) rather than functionality (all amplifiers, mixers, modulators, demodulators etc. grouped together).

The use model for *parameter-based behavioral models* is to simply set a series of parameters prior to using the model. The use model for *data-based behavioral models* is described in Chapter 8, System Data Models.

Filter Categories

The filter component libraries contains filters in eight response categories: Butterworth, Chebyshev, Elliptic, Gaussian, Bessel-Thompson, Raised-Cosine, Pole-Zero, and Polynomial. Lowpass, highpass, bandpass, and bandstop filters are available in each category.

Butterworth, Chebyshev, and Elliptic filters have good selectivity but poor group delay flatness; Bessel-Thompson and Gaussian filters have good delay flatness but poor selectivity. Raised-Cosine filters are uni-directional ideal Nyquist filters for bandlimiting digital signals. Pole-Zero and Polynomial filters allow users to define arbitrary response shapes.

Except for the Raised-Cosine category, the filter S-parameters are calculated based on standard filter synthesis theory [1]. S_{21} and S_{12} include losses specified by unloaded quality factor (Q_U) and insertion loss IL, where applicable. The assumption is made that filter pole predistortion is used to preserve the specified frequency response in the presence of losses [2]. However, S_{11} and S_{22} neglect losses, an approximation that causes little error for realizable filters.

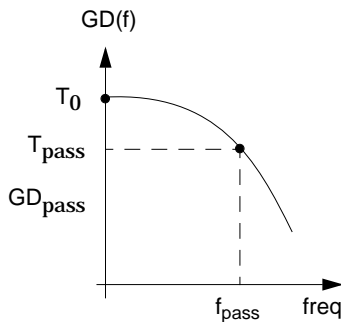
The basic nature of the response of a lowpass, highpass, bandpass, and bandstop filter is illustrated below. The illustrations include certain filter parameter definitions: f_{pass} , f_{stop} , A_{pass} , and A_{stop} for lowpass/highpass filters, and f_{center} , BW_{pass} , BW_{stop} , A_{pass} , and A_{stop} for bandpass/bandstop filters. (Note that all filters do not have all of these parameters.)

In addition to these filter parameters, Gaussian and Bessel filters (XXX_Gaussian and XXX_Bessel (XXX = LPF, HPF, BPF, or BSF) have a group delay parameter GD_{pass} . This parameter is motivated by the fact that a signal experiences a delay when passing through a filter. Calculated as the negative of the derivative of the phase response with respect to frequency, this delay (the group delay) will be frequency dependent for filters with a non-linear phase response. In other words, the group delay at a given frequency specifies the delay experienced by a group of sinusoidal components all having frequencies within a narrow interval around that frequency. A filter whose time-domain impulse response is symmetric around $t=0$ must be subjected to a certain finite shift (the group delay) in order to maintain (at least some resemblance of) causality when a transient simulator truncates its impulse response below zero. Different time-frequency characteristics for different filters can lead to different requirements for this group delay.

For LPF_Bessel and LPF_Gaussian, the group delay parameter GD_{pass} is defined as the group delay at the passband edge frequency relative to that at zero frequency (this is illustrated below). For highpass, bandpass, or bandstop filters, the definition is the same except that the group delay at infinity (highpass) or at the center frequency (bandpass, bandstop) is used as the reference. GD_{pass} cannot drop below zero and cannot exceed one; its default value is 0.9. The group delay value has a significant effect on filter order and therefore filter rejection. Larger GD_{pass} values will result in longer delays and larger filter orders; smaller GD_{pass} values will result in smaller delays and smaller filter orders. If your Gaussian or Bessel filter provides less out-of-band rejection than you expect, try increasing the GD_{pass} parameter.

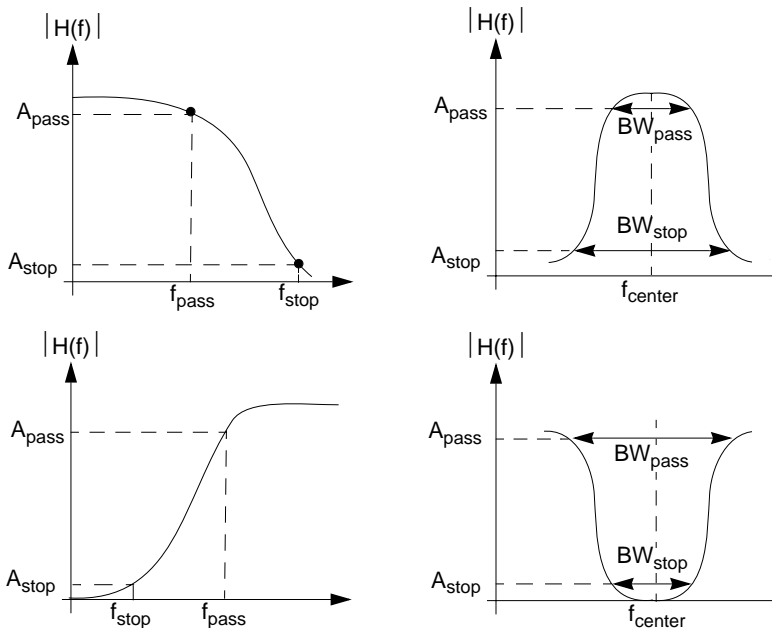
There are two ways of dealing with filter order, N:

- Leave the filter order at zero and specify the parameters characterizing the behavior of the filter, per the illustrations below. Given these specifications, the program will calculate and report filter order N which meets these specifications. Because N must be an integer, the calculated order N exceeds the specifications in most cases. N is capped at the upper value of 15. For Chebyshev filters, the calculated filter order N must be an odd number to ensure filter symmetry.
- Alternatively, specify filter order N explicitly. If a non-zero N is specified, it will overwrite the filter specifications. The filter response is simply calculated based on the specified order N. As above, N must be an integer and will be capped at 15. And, N must be odd for Chebyshev filters.



where

$$GD_{\text{pass}} = T_{\text{pass}}/T_0$$



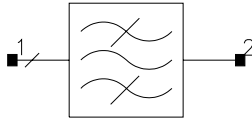
Lowpass Filter Behavior at DC

At DC, a lowpass filter appears to reduce to a wire. During DC analysis, one would therefore intuitively expect the S-matrix of a lowpass filter to reduce to $S_{11}=S_{22}=1$ and $S_{12}=S_{21}=0$. Also, one would intuitively expect the input and output voltages and currents to fulfill $v_2=v_1$ and $i_2=i_1$. For $Z_1=Z_2$, both these expectations are met. However, if Z_1 and Z_2 are unequal, there is simply no way to mathematically realize a meaningful filter that meets both expectations. If it meets the expectations in the S-domain, it won't for voltages and currents and vice versa.

The ADS lowpass filters are implemented in the S-domain, not the voltage/current-domain. At DC, the lowpass filters therefore reduce to a perfect $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ S-matrix regardless of the choice of reference impedance. The output voltage and current, however, are given by $v_2=\sqrt{Z_2/Z_1} \times v_1$ and $i_2=\sqrt{Z_1/Z_2} \times i_1$. This, as pointed out previously, reduces to $v_2=v_1$ and $i_2=i_1$ only for $Z_1=Z_2$. These relations can easily be derived from power conservation and can also be found from more rigorous S-parameter analysis.

BPF_Bessel (Bandpass Filter, Bessel-Thompson)

Symbol



Parameters

Fcenter = center frequency, in hertz

BWpass = width measured from lower to upper passband edges, in hertz

Apass = attenuation at passband edges, in dB

GDpass = group delay at passband edges relative to that at center frequency

StopType = stopband input impedance type: OPEN or SHORT

MaxRej = maximum rejection level, in dB

N = filter order; if not given, it is calculated based on BWpass, Apass, and GDpass

IL = insertion loss, in dB

Qu = unloaded quality factor for resonators, default setting is an infinite Qu and expresses a dissipationless resonant circuit.

Z1 = reference impedance for port 1, in ohms

Z2 = reference impedance for port 2, in ohms

Temp = temperature in °C

Range of Usage

$BW_{pass} < F_{center}$

$0.01 \leq A_{pass} \leq 3.0$

$0 < GD_{pass} < 1$

$1 \leq N \leq 15$

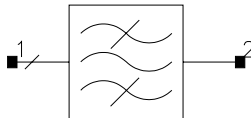
$Q_u \geq 1$

Notes/Equations

1. Refer to the section [“Filter Categories” on page 2-2](#) at the beginning of this chapter.
2. This component has no default artwork associated with it.

BPF_Butterworth (Bandpass Filter, Butterworth)

Symbol



Parameters

Fcenter = center frequency, in hertz

BWpass = width of passband, in hertz

Apass = attenuation at passband edges, in dB

BWstop = width measured from lower to upper stopband edges, in hertz

Astop = attenuation at stopband edges, in dB

StopType = stopband input impedance type: OPEN or SHORT

MaxRej = maximum rejection level, in dB

N = filter order; if not given, it is calculated based on BWpass, Apass, BWstop, and Astop

IL = insertion loss, in dB

Qu = unloaded quality factor for resonators, default setting is an infinite Qu and expresses a dissipationless resonant circuit.

Z1 = reference impedance for port 1, in ohms

Z2 = reference impedance for port 2, in ohms

Temp = temperature in °C

Range of Usage

$1 \leq N \leq 15$

$BW_{pass} < F_{center}$

$0.01 \leq A_{pass} \leq 3.0$

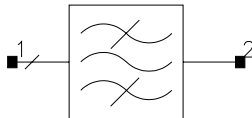
$Q_u \geq 1$

Notes/Equations

1. Refer to the section [“Filter Categories” on page 2-2](#).
2. This component has no default artwork associated with it.

BPF_Chebyshev (Bandpass Filter, Chebyshev)

Symbol



Parameters

Fcenter = center frequency, in hertz

BWpass = width of passband, in hertz

Apass = attenuation at passband edges, in dB; typically Apass=Ripple

Ripple = passband ripple, i DB

BWstop = width measured from lower to upper stopband edges, in hertz

Astop = attenuation at stopband edges, in dB

StopType = stopband input impedance type: OPEN or SHORT

MaxRej = maximum rejection level, in dB

N = filter order; if not given, is calculated based on BWpass, Ripple, BWstop, and Astop

IL = insertion loss, in dB

Qu = unloaded quality factor for resonators, default setting is an infinite Qu and expresses a dissipationless resonant circuit.

Z1 = reference impedance for port 1, in ohms

Z2 = reference impedance for port 2, in ohms

Temp = temperature in °C

Range of Usage

$BW_{pass} < F_{center}$

$0.01 \leq \text{Ripple} \leq 3.0$

$1 \leq N \leq 15$

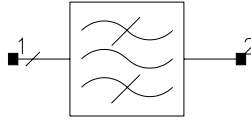
$Q_u \geq 1$

Notes/Equations

1. Refer to the section [“Filter Categories” on page 2-2](#).
2. This component has no default artwork associated with it.

BPF_Elliptic (Bandpass Filter, Elliptic)

Symbol



Parameters

Fcenter = center frequency, in hertz

BWpass = width of passband, in hertz

Ripple = passband ripple, i DB

BWstop = width measured from lower to upper stopband edges, in hertz

Astop = attenuation at stopband edges, in dB

StopType = stopband input impedance type: OPEN or SHORT

MaxRej = maximum rejection level, in dB

N = filter order; if not given, is calculated based on BWpass, Ripple, BWstop, and Astop

IL = insertion loss, in dB

Qu = unloaded quality factor for resonators, default setting is an infinite Qu and expresses a dissipationless resonant circuit.

Z1 = reference impedance for port 1, in ohms

Z2 = reference impedance for port 2, in ohms

Temp = temperature in °C

Range of Usage

$BWpass < Fcenter$

$0.01 \leq Ripple \leq 3.0$

$Astop > 0$

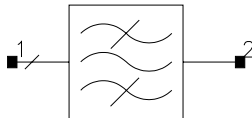
$1 \leq N \leq 15$

Notes/Equations

1. Refer to the section [“Filter Categories” on page 2-2.](#)
2. This component has no default artwork associated with it.

BPF_Gaussian (Bandpass Filter, Gaussian)

Symbol



Parameters

Fcenter = center frequency, in hertz

BWpass = width measured from lower to upper passband edges, in hertz

Apass = attenuation at passband edges, in dB

GDpass = group delay at passband edges relative to that at center frequency

StopType = stopband input impedance type: OPEN or SHORT

MaxRej = maximum rejection level, in dB

N = filter order; if not given, it is calculated based on BWpass, Apass and GDpass

IL = insertion loss, in dB

Qu = unloaded quality factor for resonators, default setting is an infinite Qu and expresses a dissipationless resonant circuit.

Z1 = reference impedance for port 1, in ohms

Z2 = reference impedance for port 2, in ohms

Temp = temperature in °C

Range of Usage

$BW_{pass} < F_{center}$

$0.01 \leq A_{pass} \leq 3.0$

$0 < GD_{pass} < 1$

$1 \leq N \leq 15$

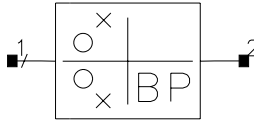
$Q_u \geq 1$

Notes/Equations

1. Refer to the section [“Filter Categories” on page 2-2.](#)
2. This component has no default artwork associated with it.

BPF_PoleZero (Bandpass Filter, Pole Zero)

Symbol



Parameters

Poles = list of poles

Zeros = list of zeros

Gain = gain factor

Fcenter = center frequency, frequency unit

BWpass = 3dB bandwidth, frequency unit

StopType = stopband input impedance type: OPEN or SHORT

Z1 = reference impedance for port 1, in ohms

Z2 = reference impedance for port 2, in ohms

Notes/Equations

1. This is an S-domain filter.

2. Poles and Zeros are a list of complex pole/zero locations.

The transfer function for the filter is:

$$S_{21} = \text{Gain} \left(\frac{(S - \text{Zero1})(S - \text{Zero2})\dots}{(S - \text{Pole1})(S - \text{Pole2})\dots} \right)$$

where

$$S = j \times \frac{F_{\text{center}}}{BW_{\text{pass}}} \left(\frac{F_{\text{req}}}{F_{\text{center}}} - \frac{F_{\text{center}}}{F_{\text{req}}} \right)$$

and

Freq is the analysis frequency

At least one pole must be supplied.

3. The following example demonstrates interpretation of simulation results with this component. From the user-specified poles/zeros, we derive:

$$S21_Lowpass_Prototype = Gain * [(s-Z1)*...*(s-Zn)] / [(s-P1)*...(s-Pm)]$$

We then check to see if $S21_Lowpass_Protope$ is > 1 . If yes, we scale $S21$ by another factor to make sure $S21_Max \leq 1$. We then derive $S11$ ($S22$) from the following formula:

$$S11^2 + S21^2 = 1$$

In this example, when Gain is set > 0.471151 , then $S11$ is derived as what you will expect. If Gain in your example is < 0.471151 , then $S11$ derived from the preceding equation, will be much higher than what you will expect. In this situation, set Gain to be 0.1 so that $S21$ has a lot of insertion loss. But we assumed there is no insertion loss in deriving $S11$.

There are other alternatives:

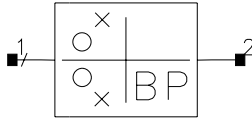
- Use $S2P_Eqn$ so that you can define the $S21$ and $S11$ polynomials however you want. You can define this as follows:

$$s=j\omega, S21 = Gain * (s-Z1)*...*(s-Zn) / (s-P1)*...*(s-Pn), S11 = \langle \text{your choice} \rangle$$

- Use BPF_Pole_Zero to model a lossless BPF, then use an attenuator to add insertion loss.

BPF_Polynomial (Bandpass Filter, Polynomial)

Symbol



Parameters

Denominator = denominator coefficients

Numerator = numerator coefficients

Gain = gain factor

Fcenter = center frequency, in hertz

BWpass = 3dB bandwidth, in hertz

StopType = stopband input impedance type: OPEN or SHORT

Z1 = reference impedance for port 1, in ohms

Z2 = reference impedance for port 2, in ohms

Notes/Equations

1. This is an S-domain filter.
2. Denominator and Numerator are a list of polynomial coefficients.

The transfer function for the filter is:

$$S_{21} = Gain \left(\frac{N_0 + (N_1 \times S) + (N_2 \times S^2) \dots}{D_0 + (D_1 \times S) + (D_2 \times S^2) \dots} \right)$$

where

$$S = j \times (\text{Freq}/F_0 - F_0/\text{Freq}) / (F_{\text{high}}/F_0 - F_0/F_{\text{high}})$$

and

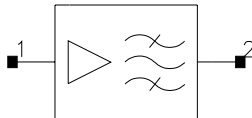
Freq is the analysis frequency

Fhigh = Fcenter + 0.5 × BWpass

Fo = sqrt((Fcenter - 0.5 × BWpass) × (Fcenter + 0.5 × BWpass))

BPF_RaisedCos (Bandpass Filter, Raised-Cosine)

Symbol



Parameters

Alpha = Rolloff factor defining filter excess bandwidth. Default = 0.35

Fcenter = Center frequency, in Hz. Default = 1.5 GHz.

SymbolRate = Digital symbol rate defining filter bandwidth, in Hz. Default = 24.3 kHz

DelaySymbols = Number of symbols delayed by filter. Default = 5

Exponent = Exponent factor, to provide for Root Raised-Cosine filter. Default = 0.5

DutyCycle = Pulse duty cycle in percent, used for sinc(x) correction. Default = 0

SincE = Flag to include the Exponent factor on the sinc(x) correction (yes/no). Default = no

Gain = Gain normalization factor. Default = 1.0

Zout = Output impedance, in Ohms. Default = 50 Ohms

WindowType = Window type applied to impulse response. 0=None (default). 1=Hann, 2=Hamming

ImpMaxFreq = Maximum frequency to consider when calculating impulse response, in Hz.

ImpDeltaFreq = Frequency sample spacing when calculating impulse response, in Hz.

ImpMaxPts = Maximum number of points in impulse response. Default = 4096

Other = output string to netlist

Range of Usage

$0 \leq \text{Alpha} \leq 1$

$\text{DelaySymbols} \geq 1$

$0 \leq \text{Exponent} \leq 1$

$0 \leq \text{DutyCycle} \leq 100$

Notes/Equations

1. Refer to the section “[Filter Categories](#)” on page 2-2.
2. This filter is unidirectional; input impedance is infinite; output impedance is specified by Z_{out} .
3. The voltage gain is described by the following function.

$$\frac{V_{out}}{V_{in}} = \frac{Gain \times G_{filt}}{G_{comp}} e^{-j2\pi frequency \left(\frac{DelaySymbols}{SymbolRate} \right)}$$

where

$$G_{filt} = 1 \text{ for } Freq < 0.5 \times (1 - Alpha) \times SymbolRate$$

$$0 \text{ for } Freq > 0.5 \times (1 - Alpha) \times SymbolRate$$

$$[0.5(1 - \sin(\pi \times (Freq - 0.5 \times SymbolRate) / (Alpha \times SymbolRate)))]^{Exponent}$$

$$G_{comp} = 1.0 \quad \text{if } DutyCycle = 0, \text{ else}$$

$$[0.01 \times DutyCycle \times \text{sinc}(x)]^{Exponent} \quad \text{if } SincE = YES, \text{ else}$$

$$[0.01 \times DutyCycle \times \text{sinc}(x)] \quad \text{if } SincE = NO$$

$$Freq = \text{abs}(Fcenter - frequency)$$

$$\text{sinc}(x) = \sin(x)/x$$

$$x = 0.01 \times DutyCycle \times \pi \times frequency / SymbolRate$$

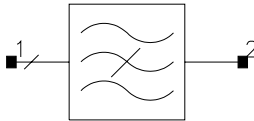
4. While the Exponent can be any value, the two standard values are 1.0 for the ideal Raised-Cosine filter response or 0.5 to simulate the Root Raised-Cosine filter response when present at both the receiving and transmitting channels.
5. In the steady-state, frequency-domain analyses, the ideal frequency-domain response described previously is used. However, this ideal response has an infinite duration impulse response which must be approximated for time domain simulations in either Transient or Circuit Envelope. If DelaySymbols parameter is set too small, then the impulse response will be severely truncated and will not accurately reflect the ideal frequency response.

Using a Hanning window (WindowType=1) a DelaySymbols parameter of 15 should result in equivalent frequency domain sidelobes of -90dBc or smaller.

6. Accuracy of this model in Transient or Circuit Envelope can be further controlled through the ImpMaxFreq, ImpDeltaFreq, and ImpMaxPts parameters.
7. The filter can include gain equalization to compensate for duty cycle roll-off. If DutyCycle = 0.0, then no compensation will be applied. If SincE=YES, Exponent will be applied to the gain compensation term, G_{comp} . (Note that the Exponent term is always present in the G_{filt} term.)
8. This component has no default artwork associated with it.

BSF_Bessel (Bandstop Filter, Bessel-Thompson)

Symbol



Parameters

Fcenter = center frequency, in hertz

BWpass = width measured from lower to upper passband edges, in hertz

Apass = attenuation at passband edges, in dB

GDpass = group delay at passband edges relative to that at center frequency

StopType = stopband input impedance type: OPEN or SHORT

Maxrej = maximum rejection level, in dB

N = filter order; if not given, it is calculated based on BWpass, Apass and GDpass

IL = insertion loss, in dB

Qu = unloaded quality factor for resonators, default setting is an infinite Qu and expresses a dissipationless resonant circuit.

Z1 = reference impedance for port 1, in ohms

Z2 = reference impedance for port 2, in ohm

Temp = temperature, in °C

Range of Usage

$BW_{pass} < F_{center}$

$0.01 \leq A_{pass} \leq 3.0$

$0 < GD_{pass} < 1$

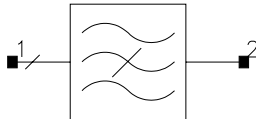
$1 \leq N \leq 15$

Notes/Equations

1. Refer to the section [“Filter Categories” on page 2-2.](#)
2. This component has no default artwork associated with it.

BSF_Butterworth (Bandstop Filter, Butterworth)

Symbol



Parameters

Fcenter = center frequency, in hertz

BWstop = width of stop band, in hertz

Astop = attenuation at stopband edges, in dB

BWpass = width measured from lower to upper passband edges, in hertz

Apass = attenuation at passband edges, in dB

StopType = stopband input impedance type: OPEN or SHORT

MaxRej = maximum rejection level, in dB

N = filter order; if not given, it is calculated based on BWpass, Apass, BWstop, Astop

IL = insertion loss, in dB

Qu = unloaded quality factor for resonators, default setting is an infinite Qu and expresses a dissipationless resonant circuit.

Z1 = reference impedance for port 1, in ohms

Z2 = reference impedance for port 2, in ohm

Temp = temperature, in °C

Range of Usage

$BW_{pass} < F_{center}$

$0.01 \leq A_{pass} \leq 3.0$

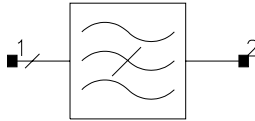
$1 \leq N \leq 15$

Notes/Equations

1. Refer to the section [“Filter Categories” on page 2-2](#).
2. This component has no default artwork associated with it.

BSF_Chebyshev (Bandstop Filter, Chebyshev)

Symbol



Parameters

Fcenter = center frequency, in hertz

BWstop = width of stop band, in hertz

Astop = attenuation at stopband edges, in dB

Ripple = stopband ripple, in dB

BWpass = width measured from lower to upper passband edges, in hertz

Apass = attenuation at passband edges, in dB

StopType = stopband input impedance type: OPEN or SHORT

MaxRej = maximum rejection level, in dB

N = filter order; if not given, it is calculated based on BWpass, Apass, BWstop, Astop

IL = insertion loss, in dB

Qu = unloaded quality factor for resonators, default setting is an infinite Qu and expresses a dissipationless resonant circuit.

Z1 = reference impedance for port 1, in ohms

Z2 = reference impedance for port 2, in ohm

Temp = temperature, in °C

Range of Usage

$BWpass < Fcenter$

$0.01 \leq Ripple \leq 3.0$

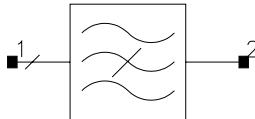
$1 \leq N \leq 15$

Notes/Equations

1. Refer to the section [“Filter Categories” on page 2-2.](#)
2. This component has no default artwork associated with it.

BSF_Elliptic (Bandstop Filter, Elliptic)

Symbol



Parameters

Fcenter = center frequency, in hertz

BWstop = width of stop band, in hertz

Astop = attenuation at stopband edges, in dB

Ripple = stopband ripple, in dB

BWpass = width measured from lower to upper passband edges, in hertz

Apass = attenuation at passband edges, in dB

StopType = stopband input impedance type: OPEN or SHORT

MaxRej = maximum rejection level, in dB

N = filter order; if not given, is calculated based on BWpass, Apass, BWstop, Astop

IL = insertion loss, in dB

Qu = unloaded quality factor for resonators, default setting is an infinite Qu and expresses a dissipationless resonant circuit.

Z1 = reference impedance for port 1, in ohms

Z2 = reference impedance for port 2, in ohm

Temp = temperature, in °C

Range of Usage

$BW_{pass} < F_{center}$

$0.01 \leq Ripple \leq 3.0$

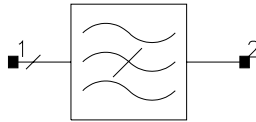
$1 \leq N \leq 15$

Notes/Equations

1. Refer to the section [“Filter Categories” on page 2-2.](#)
2. This component has no default artwork associated with it.

BSF_Gaussian (Bandstop Filter, Gaussian)

Symbol



Parameters

Fcenter = center frequency, in hertz

BWpass = width measured from lower to upper passband edges, in hertz

Apass = attenuation at passband edges, in dB

GDpass = group delay at passband edges relative to that at center frequency

StopType = stopband input impedance type: OPEN or SHORT

MaxRej = maximum rejection level, in dB

N = filter order; if not given, it is calculated based on BWpass, Apass and GDpass

IL = insertion loss, in dB

Qu = unloaded quality factor for resonators, default setting is an infinite Qu and expresses a dissipationless resonant circuit.

Z1 = reference impedance for port 1, in ohms

Z2 = reference impedance for port 2, in ohm

Temp = temperature, in °C

Range of Usage

$BW_{pass} < F_{center}$

$0.01 \leq A_{pass} \leq 3.0$

$0 < GD_{pass} < 1$

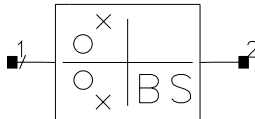
$1 \leq N \leq 15$

Notes/Equations

1. Refer to the section [“Filter Categories” on page 2-2.](#)
2. This component has no default artwork associated with it.

BSF_PoleZero (Bandstop Filter, Pole Zero)

Symbol



Parameters

Poles = list of poles

Zeros = list of zeros

Gain = gain factor

Fcenter = center frequency, in hertz

BWpass = 3dB bandwidth of lower to upper passband edges, in hertz

StopType = stopband input impedance type: OPEN or SHORT

Z1 = reference impedance for port 1, in ohms

Z2 = reference impedance for port 2, in ohms

Notes/Equations

1. This is an S-domain filter.
2. Poles and Zeros are a list of complex pole/zero locations.

The transfer function for the filter is:

$$S_{21} = Gain \left(\frac{(S - Zero1)(S - Zero2)\dots}{(S - Pole1)(S - Pole2)\dots} \right)$$

where

$$S = j \left(\frac{Fcenter}{BWpass} \right) \times \left(\frac{Freq}{Fcenter} - \frac{Fcenter}{Freq} \right)$$

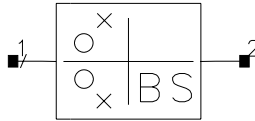
and

Freq is the analysis frequency

At least one pole must be supplied.

BSF_Polynomial (Bandstop Filter, Polynomial)

Symbol



Parameters

Denominator = list of denominator coefficients

Numerator = list of numerator coefficients

Gain = gain factor

Fcenter = center frequency, in hertz

BWpass = width of bandpass, in hertz

StopType = stopband input impedance type: OPEN or SHORT

Z1 = reference impedance for port 1, in ohms

Z2 = reference impedance for port 2, in ohms

Notes/Equations

1. This is an S-domain filter.
2. Denominator and Numerator are lists of polynomial coefficients.

The transfer function for the filter is:

$$S_{21} = Gain \left(\frac{N_0 + (N_1 \times S) + (N_2 \times S^2) \dots}{D_0 + (D_1 \times S) + (D_2 \times S^2) \dots} \right)$$

where

$$S = -j \times (F_0/F_{low} - F_{low}/F_0)/(Freq/F_0 - F_0/Freq)$$

and

Freq is the analysis frequency

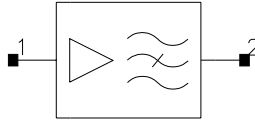
$$F_{low} = Fcenter - 0.5 \times BWpass$$

$$F_0 = \text{sqrt}((Fcenter - 0.5 \times BWpass) \times Fcenter + 0.5 \times BWpass)$$

At least one Denominator coefficient must be supplied.

BSF_RaisedCos (Bandstop Filter, Raised-Cosine)

Symbol



Parameters

Alpha = Rolloff factor defining filter excess bandwidth. Default = 0.35

Fcenter = Center frequency, in Hz. Default = 1.5 GHz.

SymbolRate = Digital symbol rate defining filter bandwidth, in Hz. Default = 24.3 KHz

DelaySymbols = Number of symbols delayed by filter. Default = 5

Exponent = Exponent factor, to provide for Root Raised-Cosine filter. Default = 0.5

DutyCycle = Pulse duty cycle in percent, used for sinc(x) correction. Default = 0

SincE = Flag to include the Exponent factor on the sinc(x) correction (yes/no). Default = no

Gain = Gain normalization factor. Default = 1.0

Zout = Output impedance, in Ohms. Default = 50 Ohms

WindowType = Window type applied to impulse response. 0=None (default). 1=Hann, 2=Hamming

ImpMaxFreq = Maximum frequency to consider when calculating impulse response, in Hz.

ImpDeltaFreq = Frequency sample spacing when calculating impulse response, in Hz.

ImpMaxPts = Maximum number of points in impulse response. Default = 4096

Other = output string to netlist

Range of Usage

$0 \leq \text{Alpha} \leq 1$

$\text{DelaySymbols} \geq 1$

$0 \leq \text{Exponent} \leq 1$

$0 \leq \text{DutyCycle} \leq 100$

Notes/Equations

1. Refer to the section “[Filter Categories](#)” on page 2-2.
2. This filter is unidirectional; input impedance is infinite; output impedance is specified by Z_{out} .
3. The voltage gain is described by the following function.

$$\frac{V_{out}}{V_{in}} = \frac{Gain \times G_{filt}}{G_{comp}} e^{-j2\pi frequency \left(\frac{DelaySymbols}{SymbolRate} \right)}$$

where

$$G_{filt} = 0 \text{ for } Freq < 0.5 \times (1 - \text{Alpha}) \times \text{SymbolRate}$$

$$1 \text{ for } Freq > 0.5 \times (1 - \text{Alpha}) \times \text{SymbolRate}$$

$$[0.5(1 + \sin(\pi \times (Freq - 0.5 \times \text{SymbolRate}) / (\text{Alpha} \times \text{SymbolRate})))]^{\text{Exponent}}$$

$$G_{comp} = 1.0 \quad \text{if DutyCycle} = 0, \text{ else}$$

$$[0.01 \times \text{DutyCycle} \times \text{sinc}(x)]^{\text{Exponent}} \quad \text{if SincE} = \text{YES}, \text{ else}$$

$$[0.01 \times \text{DutyCycle} \times \text{sinc}(x)] \quad \text{if SincE} = \text{NO}$$

$$Freq = \text{abs}(F_{center} - \text{frequency})$$

$$\text{sinc}(x) = \sin(x)/x$$

$$x = 0.01 \times \text{DutyCycle} \times \pi \times \text{frequency} / \text{SymbolRate}$$

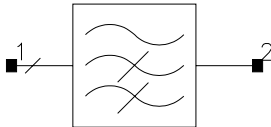
4. While the Exponent can be any value, the two standard values are 1.0 for the ideal Raised-Cosine filter response or 0.5 to simulate the Root Raised-Cosine filter response when present at both the receiving and transmitting channels.
5. In the steady-state, frequency-domain analyses, the ideal frequency-domain response described previously is used. However, this ideal response has an infinite duration impulse response which must be approximated for time domain simulations in either Transient or Circuit Envelope. If DelaySymbols parameter is set too small, then the impulse response will be severely truncated and will not accurately reflect the ideal frequency response.

Using a Hanning window (WindowType=1) a DelaySymbols parameter of 15 should result in equivalent frequency domain sidelobes of -90dBc or smaller.

6. Accuracy of this model in Transient or Circuit Envelope can be further controlled through the ImpMaxFreq, ImpDeltaFreq, and ImpMaxPts parameters.
7. The filter can include gain equalization to compensate for duty cycle roll-off. If DutyCycle = 0.0, then no compensation will be applied. If SincE=YES, Exponent will be applied to the gain compensation term, G_{comp} . (Note that the Exponent term is always present in the G_{filt} term.)
8. This component has no default artwork associated with it.

HPF_Bessel (Highpass Filter, Bessel-Thompson)

Symbol



Parameters

Fpass = passband edge frequency, in hertz

Apass = attenuation at passband edge frequency, in dB

GDpass = group delay at passband edge frequency relative to that at infinite frequency

StopType = stopband input impedance type: OPEN or SHORT

MaxRej = maximum rejection level, in dB

N = filter order; if not given, it is calculated based on Fpass, Apass and GDpass

IL = insertion loss, in dB

Qu = unloaded quality factor for resonators, default setting is an infinite Qu and expresses a dissipationless resonant circuit.

Z1 = reference impedance for port 1, in ohms

Z2 = reference impedance for port 2, in ohm

Temp = temperature, in °C

Range of Usage

$F_{pass} > 0$

$0.01 \leq A_{pass} \leq 3.0$

$0 \leq GD_{pass} < 1$

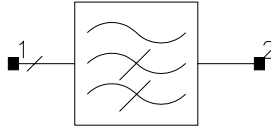
$1 \leq N \leq 15$

Notes/Equations

1. Refer to the section [“Filter Categories” on page 2-2](#).
2. This component has no default artwork associated with it.

HPF_Butterworth (Highpass Filter, Butterworth)

Symbol



Parameters

Fpass = passband edge frequency, in hertz

Apass = attenuation at passband edge frequency, in dB

Fstop = stopband edge frequency, in hertz

Astop = attenuation at stopband edge frequency, in dB

StopType = stopband input impedance type: OPEN or SHORT

MaxRej = maximum rejection level, in dB

N = filter order; if not given, it is calculated based on Fpass, Apass, Fstop, and Astop

IL = insertion loss, in dB

Qu = unloaded quality factor for resonators, default setting is an infinite Qu and expresses a dissipationless resonant circuit.

Z1 = reference impedance for port 1, in ohms

Z2 = reference impedance for port 2, in ohm

Temp = temperature, in °C

Range of Usage

$F_{pass} > 0$

$0.01 \leq A_{pass} \leq 3.0$

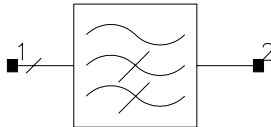
$1 \leq N \leq 15$

Notes/Equations

1. Refer to the section [“Filter Categories” on page 2-2.](#)
2. This component has no default artwork associated with it.

HPF_Chebyshev (Highpass Filter, Chebyshev)

Symbol



Parameters

Fpass = passband edge frequency, in hertz

Apass = attenuation at passband edge frequency, in dB. Typically, Apass=Ripple

Ripple = passband ripple, in dB

Fstop = stopband edge frequency, in hertz

Astop = attenuation at stopband edge frequency, in dB

StopType = stopband input impedance type: OPEN or SHORT

MaxRej = maximum rejection level, in dB

N = filter order; if not given, is calculated based on Fpass, Ripple, Fstop, and Astop

IL = insertion loss, in dB

Qu = unloaded quality factor for resonators, default setting is an infinite Qu and expresses a dissipationless resonant circuit.

Z1 = reference impedance for port 1, in ohms

Z2 = reference impedance for port 2, in ohm

Temp = temperature, in °C

Range of Usage

Fpass > 0

$0.01 \leq \text{Ripple} \leq 3.0$

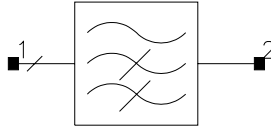
$1 \leq N \leq 15$

Notes/Equations

1. Refer to the section [“Filter Categories” on page 2-2.](#)
2. This component has no default artwork associated with it.

HPF_Elliptic (Highpass Filter, Elliptic)

Symbol



Parameters

Fpass = passband edge frequency, in hertz

Ripple = passband ripple, in dB

Fstop = stopband edge frequency, in hertz

Astop = attenuation at stopband edge frequency, in dB

StopType = stopband input impedance type: OPEN or SHORT

MaxRej = maximum rejection level, in dB

N = filter order; if not given, is calculated based on Fpass, Ripple, Fstop, and Astop

IL = insertion loss, in dB

Qu = unloaded quality factor for resonators, default setting is an infinite Qu and expresses a dissipationless resonant circuit.

Z1 = reference impedance for port 1, in ohms

Z2 = reference impedance for port 2, in ohm

Temp = temperature, in °C

Range of Usage

$F_{pass} > 0$

$0.01 \leq \text{Ripple} \leq 3.0$

$A_{stop} > 0$

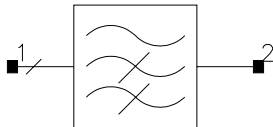
$1 \leq N \leq 15$

Notes/Equations

1. Refer to the section [“Filter Categories” on page 2-2.](#)
2. This component has no default artwork associated with it.

HPF_Gaussian (Highpass Filter, Gaussian)

Symbol



Parameters

Fpass = passband edge frequency, in hertz

Apass = attenuation at passband edge frequency, in dB

GDpass = group delay at passband edge frequency relative to that at infinite frequency

StopType = stopband input impedance type: OPEN or SHORT

MaxRej = maximum rejection level, in dB

N = filter order; if not given, it is calculated based on Fpass, Apass and GDpass

IL = insertion loss, in dB

Qu = unloaded quality factor for resonators, default setting is an infinite Qu and expresses a dissipationless resonant circuit.

Z1 = reference impedance for port 1, in ohms

Z2 = reference impedance for port 2, in ohm

Temp = temperature, in °C

Range of Usage

$F_{pass} > 0$

$0.01 \leq A_{pass} \leq 3.0$

$0 < GD_{pass} < 1$

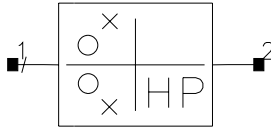
$1 \leq N \leq 15$

Notes/Equations

1. Refer to the section [“Filter Categories” on page 2-2](#).
2. This component has no default artwork associated with it.

HPF_PoleZero (Highpass Filter, Pole Zero)

Symbol



Parameters

Poles = list of poles

Zeros = list of zeros

Gain = gain factor

Fpass = 3dB passband edge frequency, in hertz

StopType = stopband input impedance type: OPEN or SHORT

Z1 = reference impedance for port 1, in ohms

Z2 = reference impedance for port 2, in ohms

Notes/Equations

1. This is an S-domain filter.
2. Poles and Zeros are a list of complex pole/zero locations.

The transfer function for the filter is:

$$S_{21} = Gain \left(\frac{(S - Zero1)(S - Zero2)...}{(S - Pole1)(S - Pole2)...} \right)$$

where

$$S = j(Fpass / Freq)$$

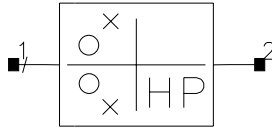
and

Freq is the analysis frequency

At least one pole must be supplied.

HPF_Polynomial (Highpass Filter, Polynomial)

Symbol



Parameters

Numerator = list of numerator coefficients

Denominator = list of denominator coefficients

Gain = gain factor

Fpass = 3dB passband edge frequency, in hertz

StopType = stopband input impedance type: OPEN or SHORT

Z1 = reference impedance for port 1, in ohms

Z2 = reference impedance for port 2, in ohms

Notes/Equations

1. This is an S-domain filter.
2. Denominator and Numerator are a list of polynomial coefficients.

The transfer function for the filter is:

$$S_{21} = \text{Gain} \left(\frac{N_0 + (N_1 \times S) + (N_2 \times S^2) \dots}{D_0 + (D_1 \times S) + (D_2 \times S^2) \dots} \right)$$

where

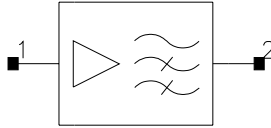
$$S = -j(\text{Fpass}/\text{Freq})$$

and

Freq is the analysis frequency

HPF_RaisedCos (Highpass Filter, Raised-Cosine)

Symbol



Parameters

Alpha = Rolloff factor defining filter excess bandwidth. Default = 0.35

SymbolRate = Digital symbol rate defining filter bandwidth, in Hz. Default = 24.3 KHz

DelaySymbols = Number of symbols delayed by filter. Default = 5

Exponent = Exponent factor, to provide for Root Raised-Cosine filter. Default = 0.5

DutyCycle = Pulse duty cycle in percent, used for sinc(x) correction. Default = 0

SincE = Flag to include the Exponent factor on the sinc(x) correction (yes/no). Default = no

Gain = Gain normalization factor. Default = 1.0

Zout = Output impedance, in Ohms. Default = 50 Ohms

WindowType = Window type applied to impulse response. 0=None (default). 1=Hann, 2=Hamming

ImpMaxFreq = Maximum frequency to consider when calculating impulse response, in Hz.

ImpDeltaFreq = Frequency sample spacing when calculating impulse response, in Hz.

ImpMaxPts = Maximum number of points in impulse response. Default = 4096

Other = output string to netlist

Range of Usage

$0 \leq \text{Alpha} \leq 1$

$\text{DelaySymbols} \geq 1$

$0 \leq \text{Exponent} \leq 1$

$0 \leq \text{DutyCycle} \leq 100$

Notes/Equations

1. Refer to the section “**Filter Categories**” on page 2-2.
2. This filter is unidirectional; input impedance is infinite; output impedance is specified by Z_{out} .
3. This filter is unidirectional; input impedance is infinite; output impedance is specified by Z_{out} .
4. The voltage gain is described by the following function.

$$\frac{V_{out}}{V_{in}} = \frac{Gain \times G_{filt}}{G_{comp}} e^{-j2\pi frequency \left(\frac{DelaySymbols}{SymbolRate} \right)}$$

where

$$G_{filt} = 0 \text{ for frequency } < 0.5 \times (1 - \text{Alpha}) \times \text{SymbolRate}$$

$$1 \text{ for frequency } > 0.5 \times (1 - \text{Alpha}) \times \text{SymbolRate}$$

$$\left[0.5(1 + \sin(\pi \times (\text{frequency} - 0.5 \times \text{SymbolRate}) / (\text{Alpha} \times \text{SymbolRate}))) \right]^{\text{Exponent}}$$

$$G_{comp} = 1.0 \quad \text{if DutyCycle} = 0, \text{ else}$$

$$\left[0.01 \times \text{DutyCycle} \times \text{sinc}(x) \right]^{\text{Exponent}} \quad \text{if SincE} = \text{YES}, \text{ else}$$

$$\left[0.01 \times \text{DutyCycle} \times \text{sinc}(x) \right] \quad \text{if SincE} = \text{NO}$$

$$\text{Freq} = \text{abs}(\text{Fcenter} - \text{frequency})$$

$$\text{sinc}(x) = \sin(x)/x$$

$$x = 0.01 \times \text{DutyCycle} \times \pi \times \text{frequency} / \text{SymbolRate}$$

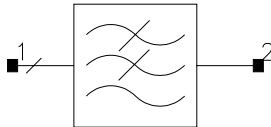
5. While the Exponent can be any value, the two standard values are 1.0 for the ideal Raised-Cosine filter response or 0.5 to simulate the Root Raised-Cosine filter response when present at both the receiving and transmitting channels.
6. In the steady-state, frequency-domain analyses, the ideal frequency-domain response described previously is used. However, this ideal response has an infinite duration impulse response which must be approximated for time domain simulations in either Transient or Circuit Envelope. If DelaySymbols parameter is set too small, then the impulse response will be severely truncated and will not accurately reflect the ideal frequency response.

Using a Hanning window (WindowType=1) a DelaySymbols parameter of 15 should result in equivalent frequency domain sidelobes of -90dBc or smaller.

7. Accuracy of this model in Transient or Circuit Envelope can be further controlled through the ImpMaxFreq, ImpDeltaFreq, and ImpMaxPts parameters.
8. The filter can include gain equalization to compensate for duty cycle roll-off. If DutyCycle = 0.0, then no compensation will be applied. If SincE=YES, Exponent will be applied to the gain compensation term, G_{comp} . (Note that the Exponent term is always present in the G_{filt} term.)
9. This component has no default artwork associated with it.

LPF_Bessel (Lowpass Filter, Bessel-Thompson)

Symbol



Parameters

Fpass = passband edge frequency, in hertz

Apass = attenuation at passband edge frequency, in dB

GDpass = group delay at passband edge frequency relative to that at zero frequency

StopType = stopband input impedance type: OPEN or SHORT

MaxRej = maximum rejection level, in dB

N = filter order; if not given, it is calculated based on Fpass, Apass, and GDpass

IL = passband insertion loss, in dB

Qu = unloaded quality factor for resonators, default setting is an infinite Qu and expresses a dissipationless resonant circuit.

Z1 = reference impedance for port 1, in ohms

Z2 = reference impedance for port 2, in ohm

Temp = temperature, in °C

Range of Usage

$F_{pass} > 0$

$0.01 \leq A_{pass} \leq 3.0$

$0 < GD_{pass} < 1$

$1 \leq N \leq 15$

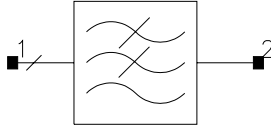
$Q_u \geq 1$

Notes/Equations

1. Refer to the section [“Filter Categories” on page 2-2.](#)
2. For information on LPF behavior at DC, refer to [“Lowpass Filter Behavior at DC” on page 2-4.](#)
3. This component has no default artwork associated with it.

LPF_Butterworth (Lowpass Filter, Butterworth)

Symbol



Parameters

Fpass = passband edge frequency, in hertz

Apass = attenuation at passband edge, in dB

Fstop = stopband edge frequency, in hertz

Astop = attenuation at stopband edge, in dB

StopType = stopband input impedance type: OPEN or SHORT

MaxRej = maximum rejection level, in dB

N = filter order (if $N > 0$, it overwrites Fstop and Astop)

IL = passband insertion loss, in dB

Qu = unloaded quality factor for resonators, default setting is an infinite Qu and expresses a dissipationless resonant circuit.

Z1 = reference impedance for port 1, in ohms

Z2 = reference impedance for port 2, in ohm

Temp = temperature, in °C

Range of Usage

$F_{pass} > 0$

$0.01 \leq A_{pass} \leq 3.0$

$1 \leq N \leq 15$

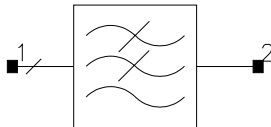
$Q_u \geq 1$

Notes/Equations

1. Refer to the section [“Filter Categories” on page 2-2.](#)
2. For information on LPF behavior at DC, refer to [“Lowpass Filter Behavior at DC” on page 2-4.](#)
3. This component has no default artwork associated with it.

LPF_Chebyshev (Lowpass Filter, Chebyshev)

Symbol



Parameters

Fpass = passband edge frequency, in hertz

Apass = attenuation at passband edge, in dB. By default, Apass=Ripple

Ripple = passband ripple, in dB

Fstop = stopband edge frequency, in hertz

Astop = attenuation at stopband edge, in dB

StopType = stopband input impedance type: OPEN or SHORT

MaxRej = maximum rejection level, in dB

N = filter order (if $N > 0$, it overwrites Fstop, and Astop)

IL = passband insertion loss, in dB

Qu = unloaded quality factor for resonators, default setting is an infinite Qu and expresses a dissipationless resonant circuit.

Z1 = reference impedance for port 1, in ohms

Z2 = reference impedance for port 2, in ohm

Temp = temperature, in °C

Range of Usage

Fpass > 0

$0.01 \leq \text{Ripple} \leq 3.0$

$1 \leq N \leq 15$

Qu ≥ 1

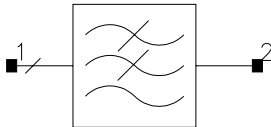
Notes/Equations

1. Refer to the section [“Filter Categories”](#) on page 2-2.
2. For information on LPF behavior at DC, refer to [“Lowpass Filter Behavior at DC”](#) on page 2-4.

3. This component has no default artwork associated with it.

LPF_Elliptic (Lowpass Filter, Elliptic)

Symbol



Parameters

Fpass = passband edge frequency, in hertz

Ripple = passband ripple, in dB

Fstop = stopband edge frequency, in hertz

Astop = attenuation at stopband edge, in dB

StopType = stopband input impedance type: OPEN or SHORT

MaxRej = maximum rejection level, in dB

N = filter order (if $N > 0$, it overwrites Fstop, and Astop)

IL = passband insertion loss, in dB

Qu = unloaded quality factor for resonators, default setting is an infinite Qu and expresses a dissipationless resonant circuit.

Z1 = reference impedance for port 1, in ohms

Z2 = reference impedance for port 2, in ohm

Temp = temperature, in °C

Range of Usage

$F_{pass} > 0$

$0.01 \leq \text{Ripple} \leq 3.0$

Astop > 0

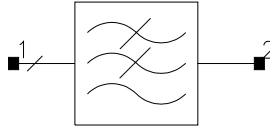
$1 \leq N \leq 15$

Notes/Equations

1. Refer to the section [“Filter Categories” on page 2-2.](#)
2. For information on LPF behavior at DC, refer to [“Lowpass Filter Behavior at DC” on page 2-4.](#)
3. This component has no default artwork associated with it.

LPF_Gaussian (Lowpass Filter, Gaussian)

Symbol



Parameters

Fpass = passband edge frequency, in hertz

Apass = attenuation at passband edge frequency, in dB

GDpass = group delay at passband edge frequency relative to that at zero frequency

StopType = stopband input impedance type: OPEN or SHORT

MaxRej = maximum rejection level, in dB

N = filter order; if not given, it is calculated based on Fpass, Apass, and GDpass

IL = passband insertion loss, in dB

Qu = unloaded quality factor for resonators, default setting is an infinite Qu and expresses a dissipationless resonant circuit.

Z1 = reference impedance for port 1, in ohms

Z2 = reference impedance for port 2, in ohm

Temp = temperature, in °C

Range of Usage

$F_{pass} > 0$

$0.01 \leq A_{pass} \leq 3.0$

$0 < GD_{pass} < 1$

$1 \leq N \leq 15$

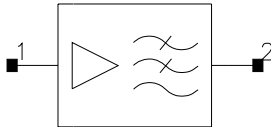
$Q_u \geq 1$

Notes/Equations

1. Refer to the section [“Filter Categories” on page 2-2](#).
2. For information on LPF behavior at DC, refer to [“Lowpass Filter Behavior at DC” on page 2-4](#).
3. This component has no default artwork associated with it.

LPF_GMSK (Lowpass Filter, GMSK)

Symbol



Parameters

BT = Product of 3dB bandwidth and bit duration

BitRate = Digital bit rate defining filter bandwidth, in Hz. Default = 270.833 KHz

DelayBits = Number of bits delayed by filter. Default = 5

Gain = Gain normalization factor. Default = 1.0

Zout = Output impedance, in Ohms. Default = 50 Ohms

WindowType = Window type applied to impulse response. 0=None (default), 1=Hann, 2=Hamming

ImpMaxFreq = Maximum frequency to consider when calculating impulse response, in Hz.

ImpDeltaFreq = Frequency sample spacing when calculating impulse response, in Hz.

ImpMaxPts = Maximum number of points in impulse response. Default = 4096

Range of Usage

$0 \leq \text{Alpha} \leq 1$

$\text{DelayBits} \geq 1$

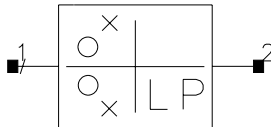
Notes/Equations/Reference

1. Refer to the section “[Filter Categories](#)” on page 2-2.
2. For information on lowpass filter behavior at DC, refer to “[Lowpass Filter Behavior at DC](#)” on page 2-4.
3. This filter is unidirectional. Its input impedance is infinite and its output impedance is specified by Zout.
4. This Gaussian filter is used in the GMSK modulation system.

5. In the steady-state frequency domain analysis, an ideal frequency-domain response is used. However, this ideal response has an infinite duration impulse response that must be approximated for time domain simulations in either Transient or Circuit Envelope. If DelayBits is set too small, then the impulse response will be severely truncated and will not accurately reflect the ideal frequency response.
6. A value of 1.0 will be used internally when DelayBits is set to a value that is less than 1.0.
7. Accuracy of this model in Transient or Circuit Envelope can be further controlled through the ImpMaxFreq, ImpDeltaFreq, and ImpMaxPts parameters.
8. This component has no default artwork associated with it.

LPF_PoleZero (Lowpass Filter, Pole Zero)

Symbol



Parameters

Poles = list of poles

Zeros = list of zeros

Gain = gain factor

Fpass = 3dB bandwidth

StopType = stopband input impedance type: OPEN or SHORT

Z1 = reference impedance for port 1

Z2 = reference impedance for port 2

Notes/Equations

1. This is an S-domain filter.
2. Poles and Zeros are a list of complex pole/zero locations.

The transfer function for the filter is:

$$S_{21} = Gain \left(\frac{(S - Zero1)(S - Zero2)...}{(S - Pole1)(S - Pole2)...} \right)$$

where

$$S = j(\text{Freq} / \text{Fpass})$$

and

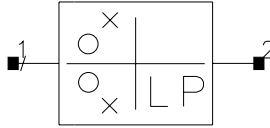
Freq is the analysis frequency

At least one pole must be supplied.

3. For information on lowpass filter behavior at DC, refer to [“Lowpass Filter Behavior at DC” on page 2-4](#).

LPF_Polynomial (Lowpass Filter, Polynomial)

Symbol



Parameters

Numerator = list of numerator coefficients

Denominator = list of denominator coefficients

Gain = gain factor

Fpass = 3dB bandwidth

StopType = stopband input impedance type: OPEN or SHORT

Z1 = reference impedance for port 1, in ohms

Z2 = reference impedance for port 2, in ohms

Notes/Equations

1. This is an S-domain filter.
2. Denominator and Numerator are a list of polynomial coefficients.

The transfer function for the filter is:

$$S_{21} = Gain \left(\frac{N_0 + (N_1 \times S) + (N_2 \times S^2) \dots}{D_0 + (D_1 \times S) + (D_2 \times S^2) \dots} \right)$$

where

$$S = j(\text{Freq} / \text{Fpass})$$

and

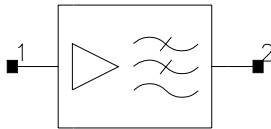
Freq is the analysis frequency

At least one denominator coefficient must be supplied.

3. For information on lowpass filter behavior at DC, refer to [“Lowpass Filter Behavior at DC” on page 2-4.](#)

LPF_RaisedCos (Lowpass Filter, Raised-Cosine)

Symbol



Parameters

Alpha = Rolloff factor defining filter excess bandwidth. Default = 0.35

SymbolRate = Digital symbol rate defining filter bandwidth, in Hz. Default = 24.3 KHz

DelaySymbols = Number of symbols delayed by filter. Default = 5

Exponent = Exponent factor, to provide for Root Raised-Cosine filter. Default = 0.5

DutyCycle = Pulse duty cycle in percent, used for sinc(x) correction. Default = 0

SincE = Flag to include the Exponent factor on the sinc(x) correction (yes/no). Default = no

Gain = Gain normalization factor. Default = 1.0

Zout = Output impedance, in Ohms. Default = 50 Ohms

WindowType = Window type applied to impulse response. 0=None (default). 1=Hann, 2=Hamming

ImpMaxFreq = Maximum frequency to consider when calculating impulse response, in Hz.

ImpDeltaFreq = Frequency sample spacing when calculating impulse response, in Hz.

ImpMaxPts = Maximum number of points in impulse response. Default = 4096

Other = output string to netlist

Range of Usage

$0 \leq \text{Alpha} \leq 1$

$\text{DelaySymbols} \geq 1$

$0 \leq \text{Exponent} \leq 1$

$0 \leq \text{DutyCycle} \leq 100$

Notes/Equations

1. Refer to the section “[Filter Categories](#)” on page 2-2.
2. For information on lowpass filter behavior at DC, refer to “[Lowpass Filter Behavior at DC](#)” on page 2-4.
3. This filter is unidirectional; input impedance is infinite; output impedance is specified by Z_{out} .
4. The voltage gain is described by the following function.

$$\frac{V_{out}}{V_{in}} = \frac{Gain \times G_{filt}}{G_{comp}} e^{-j2\pi frequency \left(\frac{DelaySymbols}{SymbolRate} \right)}$$

where

$$G_{filt} = 1 \text{ for frequency } < 0.5 \times (1 - \text{Alpha}) \times \text{SymbolRate}$$

$$0 \text{ for frequency } > 0.5 \times (1 - \text{Alpha}) \times \text{SymbolRate}$$

$$[0.5(1 - \sin(\pi \times (\text{frequency} - 0.5 \times \text{SymbolRate}) / (\text{Alpha} \times \text{SymbolRate})))]^{\text{Exponent}}$$

$$G_{comp} = 1.0 \quad \text{if DutyCycle} = 0, \text{ else}$$

$$[0.01 \times \text{DutyCycle} \times \text{sinc}(x)]^{\text{Exponent}} \quad \text{if SincE} = \text{YES}, \text{ else}$$

$$[0.01 \times \text{DutyCycle} \times \text{sinc}(x)] \quad \text{if SincE} = \text{NO}$$

$$\text{Freq} = \text{abs}(\text{Fcenter} - \text{frequency})$$

$$\text{sinc}(x) = \frac{\sin(x)}{x}$$

$$x = 0.01 \times \text{DutyCycle} \times \pi \times \text{frequency} / \text{SymbolRate}$$

5. While the Exponent can be any value, the two standard values are 1.0 for the ideal Raised-Cosine filter response or 0.5 to simulate the Root Raised-Cosine filter response when present at both the receiving and transmitting channels.
6. In the steady-state, frequency-domain analyses, the ideal frequency-domain response described previously is used. However, this ideal response has an infinite duration impulse response which must be approximated for time domain simulations in either Transient or Circuit Envelope. If DelaySymbols parameter is set too small, then the impulse response will be severely truncated and will not accurately reflect the ideal frequency response.

Using a Hanning window (WindowType=1) a DelaySymbols parameter of 15 should result in equivalent frequency domain sidelobes of -90dBc or smaller.

7. Accuracy of this model in Transient or Circuit Envelope can be further controlled through the ImpMaxFreq, ImpDeltaFreq, and ImpMaxPts parameters.
8. The filter can include gain equalization to compensate for duty cycle roll-off. If DutyCycle = 0.0, then no compensation will be applied. If SincE=YES, Exponent will be applied to the gain compensation term, G_{comp} . (Note that the Exponent term is always present in the G_{filt} term.)
9. This component has no default artwork associated with it.

SAW_Filter (Saw Filter)

Symbol



Parameters

Fcenter = passband center frequency

IL = passband insertion loss, in dB

BWpass = passband edge-to-edge width

Apass = attenuation at passband edges, in dB

BWstop = stopband edge-to-edge width

Astop = attenuation at stopband edges, in dB

GDdelay = group delay

Aripple = passband amplitude ripple, in dB

PhaRipple = passband phase ripple from linear phase, in degrees

MaxRej = maximum rejection level, in dB

Zin = input impedance

Zout = output impedance

WindowType = Window Type, 0 = None, 1 = Hann, 2 = Hamming

Temp = temperature, in °C

Other = output string to netlist

Range of Usage

$BW_{stop} \geq BW_{pass}$

$GD_{delay} \geq \pi/BW_{stop}$

Notes/Equations

1. SAW_Filter models the typical behavior of SAW bandpass filters. If the PhaRipple parameter is not specified, the filter will have perfect linear phase.
2. To maintain causality, Gdelay must be set to at least π/BW_{stop} .

References

- [1] A. I. Zverev, *Handbook of Filter Synthesis*, John Wiley and Sons, New York, 1967.
- [2] H. J. Blinchikoff and A. I. Zverev, *Filtering in the Time and Frequency Domains*, John Wiley and Sons, New York, 1976.
- [3] Kamilo Feher, *Digital Communications: Microwave Applications*, Prentice-Hall, Englewood Cliffs, N.J., 1981.
- [4] John G. Proakis, *Digital Communications*, McGraw-Hill, 1989.

Chapter 3: Modulators and Demodulators

Introduction

The *Filters* - *<filter type>* and *System* - *<device type>* palettes contain two fundamentally different types of behavioral system models.

Filters, *System - Amps & Mixers*, and *System - Mod/Demod* can be classified as *tops-down system models* that support a tops-down system design flow where model behaviors are characterized by a small number of independent parameters such as frequency, power and load. They are often referred to as *parameter-based behavioral models*.

System - Data Models can be classified as *bottoms-up system models* that support a bottoms-up verification flow where model behaviors are extracted from a simulation (or measurement) of a transistor-level circuit. They are often referred to as *data-based behavioral models*.

The *parameter-based behavioral models* typically provide superior speed relative to the data-based behavioral models with both of these being vastly superior to a brute-force transistor-level simulation.

The *data-based behavioral models* typically provide superior accuracy relative to the *parameter-based behavioral models* as they capture actual behaviors of implemented circuit components and not just design specifications.

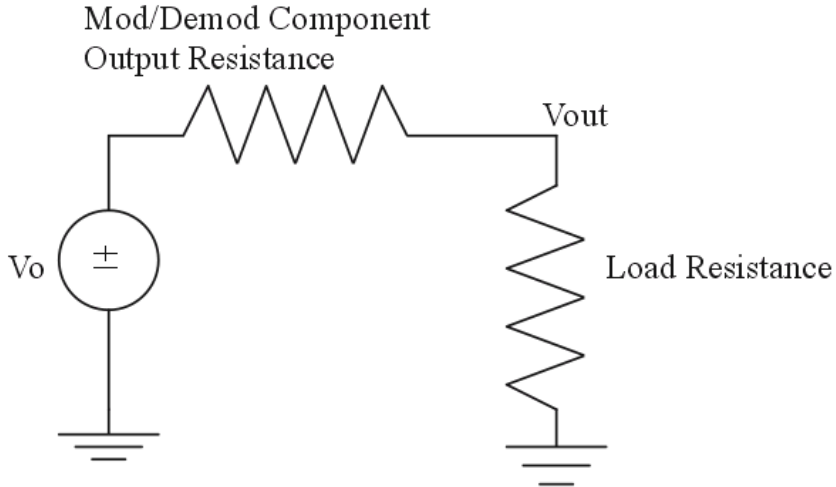
The differences between *parameter-* and *data-based behavioral models* justify a palette emphasis on flow (all data-based behavioral models grouped together) rather than functionality (all amplifiers, mixers, modulators, and demodulators grouped together) and resulted in the addition of a *System - Data Models* palette.

The use model for parameter-based behavioral models is to simply set a series of parameters prior to using the model. The use model for data-based behavioral models is slightly more involved. For a discussion, refer to Chapter 8, System Data Models.

The modulators and demodulators system model library contains time domain tuned modulators and tuned demodulators. Each component in this library is described following this introduction.

When using the System-Mod/Demod components in the Analog/RF schematic, it is important to note that terminating the component with a load resistance equivalent to the component output resistance will provide an output voltage that is half of the

applied input voltage. Consider the Thevenin equivalent of the output of a Mod/Demod component.

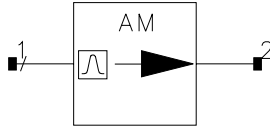


In an Analog/RF schematic, the value for V_{out} will be $1/2 V_o$ when Output Resistance=Load Resistance. In general, $V_{out}=V_o \times \text{Load Resistance}/(\text{Load Resistance} + \text{Output Resistance})$. Thus, this is the potential divider action. All of the components in the System-Mod/Demod library have this property except for N_StateDemod and PM_UnwrapDemodTuned. (These two components do not have an Rout parameter).

For a similar circuit in a DSP schematic, the effect of the potential divider on the output voltage will not be noticeable when the output resistance equals the load resistance. In this case, V_{out} equals V_o . For the DSP components, there is an additional factor of 2 at the output voltage to cancel the factor of $1/2$ from the potential divider. For a description of the digital implementation of the modulators and demodulators, and the potential divider action, refer to *Introduction: Timed, Modem Components* in the Signal Processing component documentation.

AM_DemodTuned (AM Demodulator, Tuned)

Symbol



Parameters

Fnom = nominal input frequency, in hertz

Rout = output resistance, in ohms

Notes/Equations

1. This is a tuned demodulator that selects the input harmonic closest to the specified Fnom frequency and generates a baseband output signal equal to the instantaneous amplitude of the selected carrier frequency.

If there is no analysis harmonic frequency close enough to the Fnom frequency, a warning is issued and the output is 0. Input impedance is infinite; output impedance is set by Rout, and is limited to a minimum value of 0.1 ohm. The open circuit output voltage is defined by

$$Vout_0 = mag(Vin_k)$$

where

$$mag(V) = \sqrt{Re(V)^2 + Im(V)^2}$$

and

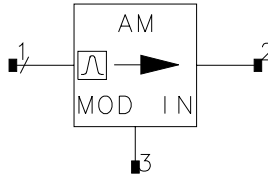
k represents the value at the analysis harmonic frequency closest to the Fnom value

0 represents the baseband component of the output voltage. All non-baseband output frequency components are 0.

2. This model generates only the absolute value of the input because the imaginary part of baseband signals is 0 for transient and baseband envelope simulations.

AM_ModTuned (AM Modulator, Tuned)

Symbol



Parameters

ModIndex = modulation index

Fnom = nominal input frequency, in hertz

Rout = output resistance, in ohms

Notes/Equations

1. This is a tuned modulator that selects the input harmonic defined by the specified Fnom frequency and amplitude modulates it by the baseband modulation input.

If there is no analysis harmonic frequency close enough to the Fnom frequency, a warning is issued and the output is 0. The two input impedances are infinite. The output impedance is set by Rout, and is limited to a minimum value of 0.1 ohm. The open circuit output voltage is defined by

$$Vout_k = (1 + ModIndex \times Vmod_0) \times Vin_k$$

where

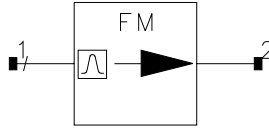
k represents the value at the analysis harmonic frequency closest to the Fnom value

0 represents the baseband component of the modulation input voltage.

2. The RF carrier is injected at pin 1; the modulating signal is injected at pin 3. The resulting AM signal is present at pin 2. The RF carrier should be a frequency-domain source; the modulating signal should be a time-domain source.
3. This model works in transient and baseband envelope simulations. Because the describing equation is valid for baseband signals, Fnom has no effect.

FM_DemodTuned (FM Demodulator, Tuned)

Symbol



Parameters

Sensitivity = demodulation sensitivity, in hertz/volt

Fnom = nominal input frequency, in hertz

Rout = output resistance, in ohms

Notes/Equations

1. This is a tuned demodulator that selects the input harmonic closest to the specified Fnom frequency and generates a baseband output signal equal to the instantaneous frequency of the selected carrier frequency.

If there is no analysis harmonic frequency close enough to the Fnom frequency, a warning is issued and the output is 0. Input impedance is infinite; output impedance is set by Rout, and is limited to a minimum value of 0.1 ohm.

The instantaneous frequency difference is equal to the time derivative of the instantaneous phase. This is approximated using a simple finite difference. As long as the phase does not change by greater than 180° in one time step, the frequency calculation is non-ambiguous. The open circuit output voltage can be described as:

$$Vout_0 = \frac{\Phi(Vin_k(t)) - \Phi(Vin_k(t - timestep))}{2\pi \times Sensitivity \times timestep}$$

where

$$\Phi(V) = \text{atan} \frac{\text{Im}(V)}{\text{Re}(V)}$$

k represents the value at the analysis harmonic frequency closest to the Fnom value

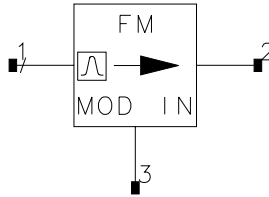
0 represents the baseband component of the output voltage.

All non-baseband output frequency components are 0.

2. For transient and baseband envelope simulations, this model does not generate the expected results because the imaginary part of baseband signals is 0.

FM_ModTuned (FM Modulator, Tuned)

Symbol



Parameters

Sensitivity = modulation sensitivity, in hertz/volt

Fnom = nominal input frequency, in hertz

Rout = output resistance, in ohms

Notes/Equations

1. This is a tuned modulator that selects the input harmonic defined by the specified Fnom frequency and frequency modulates it by the baseband modulation input.

If there is no analysis harmonic frequency close enough to the Fnom frequency, a warning is issued and the output is 0. The two input impedances are infinite. The output impedance is set by Rout, and is limited to a minimum value of 0.1 ohm. The open circuit output voltage is defined by

$$Vout_k = e^{j2\pi Sensitivity \int_0^t Vm_0 dt} \times Vin_k$$

where

k represents the value at the analysis harmonic frequency closest to the Fnom value

0 represents the baseband component of the modulation input voltage.

For time $t=0$, the integrator of the modulation input signal is reset to 0. In all steady-state analyses such as harmonic balance, dc, and ac, time is kept at 0, so no modulation occurs.

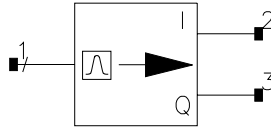
2. The RF carrier is injected at pin 1; the modulating signal is injected at pin 3. The resulting FM signal is present at pin 2. The RF carrier should be a

frequency-domain source; the modulating signal should be a time-domain source.

3. This model will not function as a frequency modulator in transient and baseband envelope simulations because of the use of the complex exponential operator.

IQ_DemodTuned (I/Q Demodulator, Tuned)

Symbol



Parameters

Fnom = nominal input frequency, in hertz

Rout = output resistance, in ohms

Notes/Equations

1. This is a tuned demodulator that selects the input harmonic closest to the specified Fnom frequency and generates two baseband output signals equal to the instantaneous in-phase and quadrature-phase components of the selected carrier frequency.

If there is no analysis harmonic frequency close enough to the Fnom frequency, a warning is issued and the output is 0. Input impedance is infinite; output impedance is set by Rout, and is limited to a minimum value of 0.1 ohm. The open circuit output voltages are defined simply by

$$Vi_0 = \text{Re}(Vin_k)$$

$$Vq_0 = \text{Im}(Vin_k)$$

where

k represents the value at the analysis harmonic frequency closest to the Fnom value

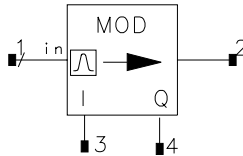
0 represents the baseband component of the output voltages.

All non-baseband output frequency components are 0.

2. In transient and baseband envelope simulations, because the imaginary part of baseband signals is 0, this model replicates the input on the I output only.

IQ_ModTuned (I/Q Modulator, Tuned)

Symbol



Parameters

Fnom = nominal input frequency, in hertz

Rout = output resistance, in ohms

Notes/Equations

1. This is a tuned modulator that selects the input harmonic defined by the specified Fnom frequency and modulates it according to the I (in-phase) and Q (quadrature) modulation inputs.

If there is no analysis harmonic frequency close enough to the Fnom frequency, a warning is issued and the output is 0. All three input impedances are infinite; output impedance is set by Rout, and is limited to a minimum value of 0.1 ohm. The open circuit output voltage is defined by

$$out_k = (V_{I0} + j \times V_{Q0}) \times Vin_k$$

where

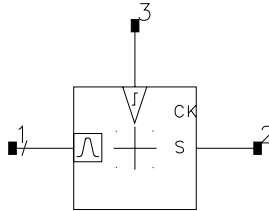
k represents the value at the analysis harmonic frequency closest to the Fnom value

0 represents the baseband component of the two modulation input voltages.

2. The RF carrier is injected at pin 1; I data is injected at pin 3; Q data is injected at pin 4. The resulting modulated signal is present at pin 2. The RF carrier should be a frequency-domain source; the modulating signals should be a time-domain source.
3. This model will not function as a phase modulator in transient and baseband envelope simulation because of the use of the complex j operator.
4. Both AM and PM modulation can be generated.

N_StateDemod (N-State Demodulator)

Symbol



Parameters

Fnom = nominal input frequency, in hertz

StateArray = complex array of state values

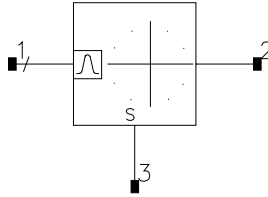
Notes/Equations

1. This is a tuned demodulator that selects the input harmonic closest to the specified Fnom frequency and generates a baseband output state signal representing the nearest complex nominal state of the selected carrier at the time of last rising clock edge.

If there is no analysis harmonic frequency close enough to the Fnom frequency, a warning is issued and the output is 0. On each rising clock edge, the I/Q information of the carrier is sampled. Using a Euclidean distance measure, the closest nominal state in the StateArray variable is determined and that state number, minus 1, is output as the baseband state voltage. This can be considered a 2-dimensional quantizer.
2. The nominal state locations are arbitrary and user definable in the StateArray variable, which can be created from a list of complex values using the list() function.
3. In transient and baseband envelope simulations, this state demodulation model functions correctly and could be used as a 1-dimensional arbitrary state quantizer.

N_StateMod (N-State Modulator)

Symbol



Parameters

MaxStates = maximum number of input states

StateArray = complex array of state values

Fnom = nominal input frequency, in hertz

Rout = output resistance, in ohms

Notes/Equations

1. This N-state modulator can be used to create an arbitrary, user-defined constellation of complex modulation states. The input harmonic closest to the specified Fnom frequency is selected and modulated by a sequence of complex states.

If there is no analysis harmonic frequency close enough to the Fnom frequency, a warning is issued and the output is 0. The states are defined by the complex array variable StateArray. The sequence of states is determined by the baseband component of the state modulation input. The two input impedances are infinite; output impedance is set by Rout, and is limited to a minimum value of 0.1 ohm. The open circuit output voltage is defined by

$$Vout_k = State[int(Vs_0) + 1] \times Vin_k$$

where

k represents the value at the analysis harmonic frequency closest to the Fnom value

0 represents the baseband component of the modulation state input voltage; this input voltage is internally limited to be between 0 and MaxStates - 1.

The StateArray variable can be created from list of complex values using the list() function.

- This model will not function as a complex modulator in transient and baseband envelope simulations if complex state values are specified.
- Figure 3-1** and **Figure 3-2** show a schematic example and simulation results for a 16-state modulator being swept linearly through all 16 of its states.

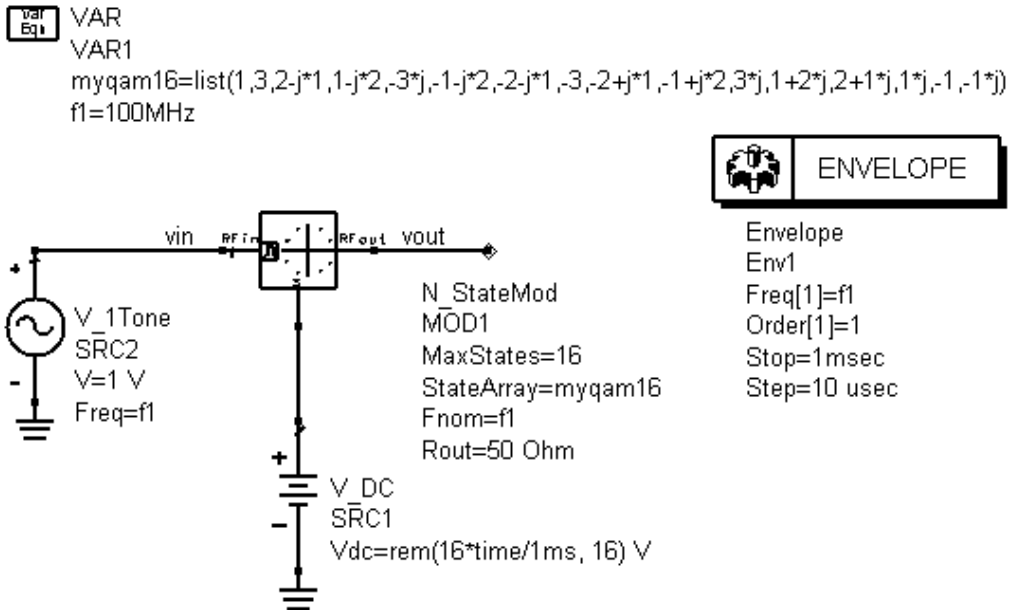


Figure 3-1. 16-State Modulation Example

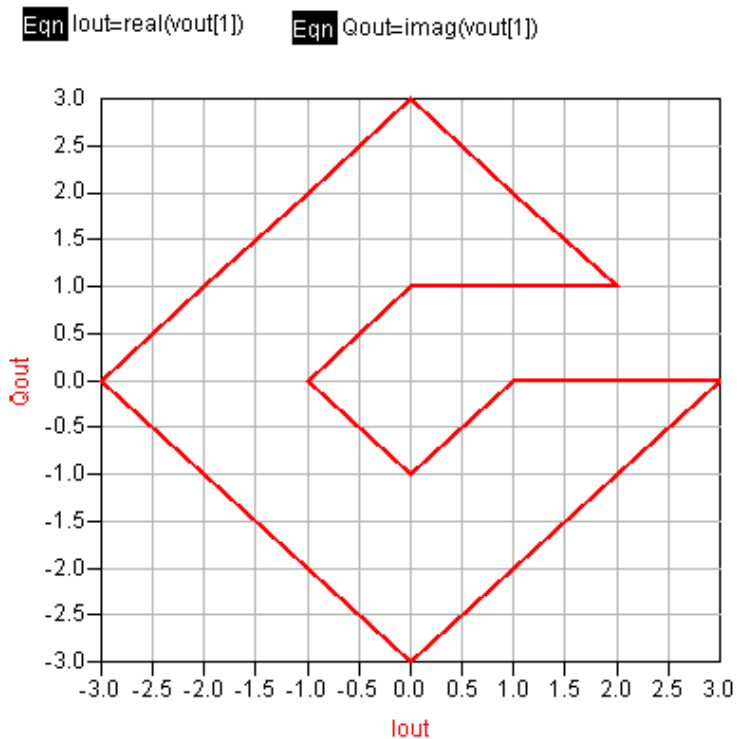
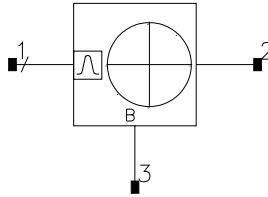


Figure 3-2. Simulation Results

PI4DQPSK_ModTuned (PI/4 DQPSK Modulator, Tuned)

Symbol



Parameters

Fnom = nominal input frequency, in hertz

Rout = output resistance, in ohms

SymbolRate = output symbol rate (one-half input bit rate)

Delay = sampling delay, in seconds

Notes/Equations

1. This tuned PI/4 DQPSK modulator selects the input harmonic closest to the specified Fnom frequency and modulates it according to the phase state determined by differentially encoding the input bit stream and applying the PI/4 phase offset.

If there is no analysis harmonic frequency close enough to the Fnom frequency, a warning is issued and the output is 0. The input bit stream is sampled at a rate determined by the SymbolRate with an initial synchronizing delay determined by the Delay parameter. The actual sampling rate is rounded to an integer multiple of the system timestep—use a corresponding discrete time source to generate this serial bit stream or ensure that the analysis sampling rate is an integer multiple of the bit rate. Two consecutive bits are monitored, along with the previous phase state, to determine the next phase state. An input bit is assumed to be 1 if it is greater than 0.5V and 0 otherwise.

2. The RF carrier is injected at pin 1; the input bit stream is injected at pin 3. The resulting PI4DQPSK signal is present at pin 2. The RF carrier should be a frequency-domain source; the modulating signal should be a time-domain source.
3. The input impedances of both the serial bit stream input and the RF carrier are infinite; output impedance is set by Rout, and is limited to a minimum value of 0.1 ohm. The open circuit output voltage is defined by:

$$Vout_k = e^{jNextPh(Vm_0(t), Vm_0(t - BitTime), CurrentPh)} \times Vin_k$$

where

k represents the value at the analysis harmonic frequency closest to the F_{nom} value

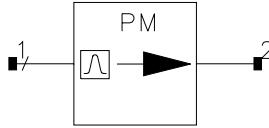
0 represents the baseband component of the bit stream input voltage.

$Vm_0(t)$	$Vm_0(t - BitTime)$	Phase Transition = Next Phase – Current Phase
0	0	45°
0	1	135°
1	0	-45°
1	1	-135°

4. This model will not function as a modulator in transient and baseband envelope simulations because of the use of complex exponential functions.

PM_DemodTuned (PM Demodulator, Tuned)

Symbol



Parameters

Sensitivity = demodulation sensitivity, in degree/volt

Fnom = nominal input frequency, in hertz

Rout = output resistance, in ohms

Notes/Equations

1. This is a tuned demodulator that selects the input harmonic closest to the specified Fnom frequency and generates a baseband output signal equal to the instantaneous phase of the selected carrier frequency.

If there is no analysis harmonic frequency close enough to the Fnom frequency, a warning is issued and the output is 0. The input impedance is infinite. The output impedance is set by Rout, and is limited to a minimum value of 0.1 ohm. The open circuit output voltage is defined by

$$Vout_0 = \frac{\Phi(Vin_k(t)) \times 180}{\pi \times Sensitivity}$$

where

$$\Phi(V) = \text{atan} \frac{\text{Im}(V)}{\text{Re}(V)}$$

k represents the value at the analysis harmonic frequency closest to the Fnom value

0 represents the baseband component of the output voltage.

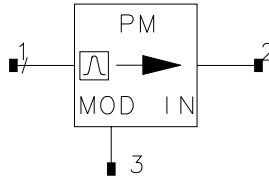
All non-baseband output frequency components are 0.

The maximum phase range of this demodulator is $\pm 180^\circ$.

2. In transient and baseband envelope simulations, this model does not generate the expected results because the imaginary part of baseband signals is 0.

PM_ModTuned (PM Modulator, Tuned)

Symbol



Parameters

Sensitivity = demodulation sensitivity, in degree/volt

Fnom = nominal input frequency, in hertz

Rout = output resistance, in ohms

Notes/Equations

1. This is a tuned modulator that selects the input harmonic defined by the specified Fnom frequency and phase modulates it by the baseband modulation input. If there is no analysis harmonic frequency close enough to the Fnom frequency, a warning is issued and the output is 0. The two input impedances are infinite. The output impedance is set by Rout, and is limited to a minimum value of 0.1 ohm. The open circuit output voltage is defined by

$$Vout_k = e^{j\pi Sensitivity \times (Vmod_0)/180} \times Vin_k$$

where

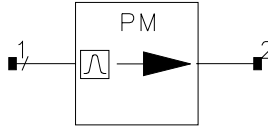
k represents the value at the analysis harmonic frequency closest to the Fnom value

0 represents the baseband component of the modulation input voltage.

2. The RF carrier is injected at pin 1; the modulating signal is injected at pin 3. The resulting PM signal is present at pin 2. The RF carrier should be a frequency-domain source; the modulating signal should be a time-domain source.
3. This model will not function as a phase modulator in transient and baseband envelope simulations because of the use of the complex exponential operator.

PM_UnwrapDemodTuned (PM Unwrapped Demodulator, Tuned)

Symbol



Parameters

Sensitivity = modulation sensitivity, in degree/volt

Fnom = nominal input frequency, in hertz

MaxAngle = unwrapped phase angle range, in degrees

Notes/Equations

1. This is a tuned demodulator that selects the input harmonic closest to the specified Fnom frequency and generates a baseband output signal equal to the instantaneous unwrapped phase of the selected carrier frequency.

If there is no analysis harmonic frequency close enough to the Fnom frequency, a warning is issued and the output is 0. The input impedance is infinite. The output impedance is 0.

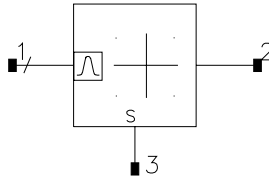
2. The phase range of this demodulator is equal to \pm MaxAngle. The phase at time 0 is set equal to the normal 180° phase of the selected carrier frequency. By tracking modulo 360° phase transitions, the unwrapped phase, relative to the initial time 0 value, is calculated. The user can then set the desired modulo range with the MaxAngle parameter. A large number will generate totally unwrapped phase.

This unwrapping of phase works for time sweeps in envelope analyses; it cannot be used to unwrap the phase of frequency or other parameter sweeps.

3. In transient and baseband envelope simulations, this model does not generate the expected results because the imaginary part of baseband signals is 0.

QPSK_ModTuned (QPSK Modulator, Tuned)

Symbol



Parameters

Fnom = nominal input frequency, in hertz

Rout = output resistance, in ohms

State1 = complex modulation coordinates of State 1

State2 = complex modulation coordinates of State 2

State3 = complex modulation coordinates of State 3

State4 = complex modulation coordinates of State 4

Notes/Equations

1. This tuned quadrature phase shift keying (QPSK) modulator is actually an arbitrary 4-state modulator. The input harmonic closest to the specified Fnom frequency is selected and modulated by one of four user-defined complex values. If there is no analysis harmonic frequency close enough to the Fnom frequency, a warning is issued and the output is 0.

The complex values are determined by the baseband component of the state modulation input. An input less than 0.5V selects State 1; less than 1.5V selects State 2; less than 2.5V selects State 3; and, any input greater than 2.5V selects State 4. The two input impedances are infinite. The output impedance is set by Rout and is limited to a minimum value of 0.1 ohm. The open circuit output voltage is defined by:

$$Vout_k = State[int(Vs_0 + 1.5)] \times Vin_k$$

where

k represents the value at the analysis harmonic frequency closest to the Fnom value

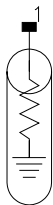
0 represents the baseband component of the modulation state input voltage.

2. The RF carrier is injected at pin 1; the input bit stream is injected at pin 3. The resulting QPSK signal is present at pin 2. The RF carrier should be a frequency-domain source; the modulating signal should be a time-domain source.
3. This model will not function as a complex modulator in transient and baseband envelope simulations if the specified state values are complex.

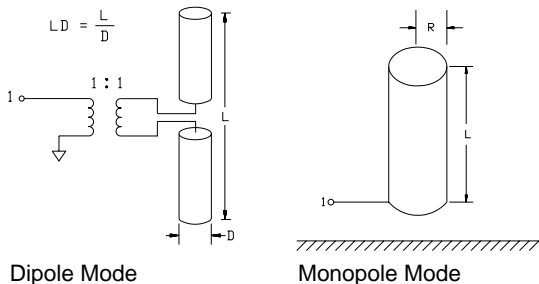
Chapter 4: Passive System Components

AntLoad (Antenna Load)

Symbol



Illustration



Parameters

AntType = antenna type: MONOPOLE or DIPOLE

Length = physical antenna length, in specified units

RatioLR = length-to-radius ratio

Range of Usage

$$\text{monopole simulation frequency (MHZ)} \leq \frac{287}{L(\text{meters})} \times \frac{\text{RatioLR}}{\text{RatioLR} + 1}$$

$$\text{dipole simulation frequency (MHZ)} \leq \frac{575}{L(\text{meters})} \times \frac{\text{RatioLR}}{\text{RatioLR} + 2}$$

Notes/Equations

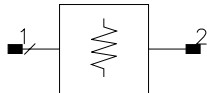
1. This component models the input impedance of a monopole or dipole antenna. Transmission is not modeled.
2. For time-domain analysis, the frequency-domain analytical model is used.
3. This component has no default artwork associated with it.

References

- [1] Constantine A. Balanis, *Antenna Theory: Analysis and Design*, Harper and Row, 1982, pp. 112, 295, 334-335.
- [2] R. I. Hansen, "Fundamental Limitations in Antennas," *Proceedings of the IEEE*, Vol. 68, No. 2, February, 1981, pp. 170-182.
- [3] John D. Kraus, *Antennas*, McGraw-Hill Book Company, Inc., 1950, pp. 276-278.

Attenuator (Attenuator)

Symbol



Parameters

Loss = attenuation, in dB

VSWR = voltage standing wave ratio for both ports

Rref = reference resistance for both ports

Temp = temperature, in °C

CheckPassivity = check passivity flag: if set to yes (default), a passivity check is performed and a warning is output if the device is not passive; if set to no, a passivity check is not performed.

ReturnPhase = phase scaling of S_{11} , S_{22} ; default is 0

Range of Usage

VSWR > 1.0

Notes/Equations

1. This component will always provide attenuation; for example, Loss=20 or Loss=-20 will both result in 20 dB attenuation.

2. S-parameters are:

$$10 \log |S_{12}|^2 = 10 \log |S_{21}|^2 = - |Loss|$$

$$S_{11} = S_{22} = \frac{VSWR - 1}{VSWR + 1}$$

$$\text{Phase}(S_{11}) = \text{Phase}(S_{22}) = -90 \text{ if ReturnPhase} < 0$$

$$\text{Phase}(S_{11}) = \text{Phase}(S_{22}) = 0 \text{ if ReturnPhase} = 0$$

$$\text{Phase}(S_{11}) = \text{Phase}(S_{22}) = 90 \text{ if ReturnPhase} > 0$$

3. When ReturnPhase=0, $|S_{11}|$ and $|S_{22}|$ will be limited to min

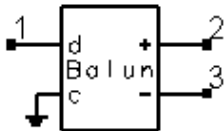
$$(|S_{11}|, 1.0 - |S_{21}|)$$

in order to maintain passivity constraints.

4. When used in time domain simulation, set ReturnPhase=0; any other ReturnPhase value will not produce proper results because (+/-)j in the frequency domain corresponds to a non-causal impulse response in time domain.

Balun3Port (Balun, 3-port)

Symbol



Parameters

None

Notes/Equations

1. Balun3Port realizes the ideal transformation between a balanced differential-mode signal and unbalanced, single-ended signals. It can be useful to connect a source to a differentially fed circuit, although it does ignore common-mode effects.
2. Balun3Port realizes the voltage and current transformations given by:

$$v_d = v_+ - v_-$$

$$i_+ = -i_- = -i_d$$

where

v_d/i_d = the differential mode voltage/current at pin d

v_+/i_+ = the single line voltage/current at pin +

v_-/i_- = the single line voltage/current at pin -

See P.R. Gray and R.G. Meyer, "Analysis and Design of Analog Integrated Circuits", Third Edition, John Wiley & Sons, NY, 1993, equations 3.91 & 3.92.

The minus signs in the current definitions are due to the standard definition of currents directed *into* the Balun3Port component.

3. An equivalent functionality can be realized with a Balun4Port that has the common-mode pin grounded. However, the Balun3Port provides better convergence properties.
4. Balun3Port is bi-directional. When fed at the differential-mode pin, it realizes the transformations:

$$v_+ = -v_- = (v_d)/2$$

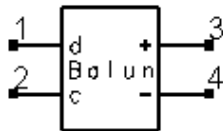
$$i_+ = -i_- = -i_d$$

5. Examples of using Balun3Port to convert between (unbalanced) ADS sources and balanced circuits can be found in the ADS examples directory; access these examples from the ADS Main window > *File* > *Example Project*.

- *RFIC* > *MixerDiffMode_prj* demonstrates the use of Balun4Port to present differential-mode sources (as well as common mode biases) to the RF and LO inputs. It also shows the use of Balun3Port to single-ended (differential-mode) IF output, which is needed to properly calculate the noise figure.
- *BehavioralModels* > *DifferentialModels_prj* demonstrates the use of Balun3Port and Balun4Port in conjunction with single-ended System-Data Models in order to create a data-based behavioral model of a differentially fed mixer.

Balun4Port (Balun, 4-port)

Symbol



Parameters

None

Notes/Equations

1. Balun4Port realizes the ideal transformation between balanced (differential- and common-mode) signals and unbalanced (single-ended) signals. It can be used to connect sources to a differentially fed circuit, particularly when modeling common-mode effects are important.
2. Balun4Port realizes voltage and current transformations given by:

$$v_d = v_+ - v_-$$

$$v_c = (v_+ + v_-)/2$$

$$i_d = -(i_+ - i_-)/2$$

$$i_c = -(i_+ + i_-)$$

where

v_d/i_d = differential mode voltage/current at pin d

v_c/i_c = common mode voltage/current at pin c

v_+/i_+ = single line voltage/current at pin +

v_-/i_- = single line voltage/current at pin -

See P.R. Gray and R.G. Meyer, "Analysis and Design of Analog Integrated Circuits", Third Edition, John Wiley & Sons, NY, 1993, equations 3.91 & 3.92.

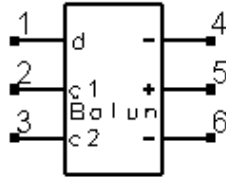
The minus signs in the current definitions are due to the standard definition of currents directed *into* the Balun4Port component.

3. Balun4Port is bi-directional. It converts common/differential-mode signals into two single-ended signals, as well as converting two single-ended signals into common/differential mode signals.

4. If common-mode effects are not desired, Balun3Port provides an equivalent, but numerically more robust, result as grounding the common-mode pin of Balun4Port.
5. Examples of using Balun4Port to convert between (unbalanced) ADS sources and balanced circuits can be found in the ADS examples directory; access these examples from the ADS Main window > *File* > *Example Project*.
 - *RFIC* > *MixerDiffMode_prj* demonstrates the use of Balun4Port to present differential-mode sources (as well as common mode biases) to the RF and LO inputs. It also shows the use of Balun3Port to single-ended (differential-mode) IF output, which is needed to properly calculate the noise figure.
 - *BehavioralModels* > *DifferentialModels_prj* demonstrates the use of Balun3Port and Balun4Port in conjunction with single-ended System-Data Models in order to create a data-based behavioral model of a differentially fed mixer.

Balun6Port (Balun, 6-port)

Symbol



Parameters

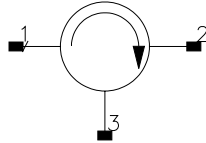
None

Notes/Equations

1. Balun6Port is based on ideal transformers; it can be used to transform a single-ended signal to a differential signal with two ground planes such as a stripline.
2. This component passes DC.

Circulator (Ideal 3-Port Circulator)

Symbol



Parameters

F1 = first frequency breakpoint

F2 = second frequency breakpoint

F3 = third frequency breakpoint

Loss1 = attenuation for frequencies $\leq F1$, in dB

Loss2 = attenuation for frequencies $>F1 \leq F2$, in dB

Loss3 = attenuation for frequencies $>F2 \leq F3$, in dB

Loss4 = attenuation for frequencies $>F3$, in dB

VSWR1 = voltage standing wave ratio at both ports for frequencies $\leq F1$

VSWR2 = voltage standing wave ratio at both ports for frequencies $>F1, \leq F2$

VSWR3 = voltage standing wave ratio at both ports for frequencies $>F2, \leq F3$

VSWR4 = voltage standing wave ratio at both ports for frequencies $>F3$

Isolat = isolation, in dB

Z1 = reference impedance for port 1

Z2 = reference impedance for port 2

Z3 = reference impedance for port 3

Temp = temperature, in degrees C

CheckPassivity = check passivity flag: if set to yes (default), a passivity check is performed and a warning is output if the device is not passive; if set to no, a passivity check is not performed.

Notes/Equations

1. Up to three frequency breakpoints can be used to define four bands. Each band will have a different Loss# and VSWR#. If no frequencies are specified, then this model is frequency independent and only uses Loss1 and VSWR1.
2. A loss is modeled regardless of the sign of the Loss parameter value.
3. All ports have the same VSWR.
4. Isolation is constant across all frequency bands.
5. The 3×3 S-matrix for Circulator is described by

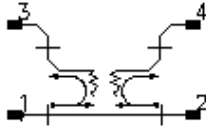
$$S_{11} = S_{22} = S_{33} = \frac{VSWR - 1}{VSWR + 1}$$

$$S_{12} = S_{23} = S_{31} = 10^{(-abs(Isolat)/20)}$$

$$S_{13} = S_{21} = S_{32} = 10^{(-abs(Loss)/20)}$$

CouplerDual (Dual Coupler)

Symbol



Parameters

Coupling = coupling factor, in dB

F1 = first frequency breakpoint

F2 = second frequency breakpoint

F3 = third frequency breakpoint

MVSWR1 = main arm VSWR for frequency \leq F1

MVSWR2 = main arm VSWR for F1 < frequency \leq F2

MVSWR3 = main arm VSWR for F2 < frequency \leq F3

MVSWR4 = main arm VSWR for frequency > F3

CVSWR1 = coupled arm VSWR for frequency \leq F1

CVSWR2 = coupled arm VSWR for F1 < frequency \leq F2

CVSWR3 = coupled arm VSWR for F2 < frequency \leq F3

CVSWR4 = coupled arm VSWR for frequency > F3

Loss1 = attenuation for frequency \leq F1, in dB

Loss2 = attenuation for F1 < frequency \leq F2, in dB

Loss3 = attenuation for F2 < frequency \leq F3, in dB

Loss4 = attenuation for frequency > F3, in dB

Direct1 = directivity for frequency \leq F1, in dB

Direct2 = directivity for F1 < frequency \leq F2, in dB

Direct3 = directivity for F2 < frequency \leq F3, in dB

Direct4 = directivity for frequency > F3, in dB

ZRef = reference impedance for all ports

Temp = temperature, in degrees C

CheckPassivity = check passivity flag: if set to yes (default), a passivity check is performed and a warning is output if the device is not passive; if set to no, a passivity check is not performed.

Notes/Equations

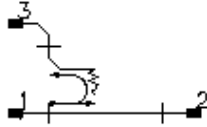
1. Port 1 is the input port; port 2 is the through port; port 3 is the coupled port; port 4 is the isolated port.
2. Both ports of the main arm (ports 1 and 2) are assumed to have the same VSWR. Both ports of the coupled arm (ports 3 and 4) are also assumed to have the same VSWR.
3. *Coupling* is assumed to be constant across all frequency bands.
4. *Loss* is the dissipation of the coupler; it affects S21 and S12 only. These S-parameters are calculated as if the coupler was dissipationless and are then scaled by

$$10^{(-abs(Loss)/20)}$$

5. Internally, coupler isolation is calculated as *Coupling + Direct* in dB.
6. Up to three frequency breakpoints can be used to define four bands, with different directivity losses and VSWR values for each frequency band. If no frequencies are specified, then this model is frequency independent and uses Direct1, Loss1, and VSWR1 only.
7. There is a 90-degree phase shift between the input and the coupled ports.
8. In the coupled arm, the two ports are uncoupled.

CouplerSingle (Single Coupler)

Symbol



Parameters

Coupling = coupling factor, in dB

F1 = first frequency breakpoint

F2 = second frequency breakpoint

F3 = third frequency breakpoint

MVSWR1 = main arm VSWR for frequency \leq F1

MVSWR2 = main arm VSWR for $F1 <$ frequency \leq F2

MVSWR3 = main arm VSWR for $F2 <$ frequency \leq F3

MVSWR4 = main arm VSWR for frequency $>$ F3

CVSWR1 = coupled arm VSWR for frequency \leq F1

CVSWR2 = coupled arm VSWR for $F1 <$ frequency \leq F2

CVSWR3 = coupled arm VSWR for $F2 <$ frequency \leq F3

CVSWR4 = coupled arm VSWR for frequency $>$ F3

Loss1 = attenuation for for frequency \leq F1, in dB

Loss2 = attenuation for $F1 <$ frequency \leq F2, in dB

Loss3 = attenuation for $F2 <$ frequency \leq F3, in dB

Loss4 = attenuation for frequency $>$ F3, in dB

Direct1 = directivity for frequency \leq F1, in dB

Direct2 = directivity for $F1 <$ frequency \leq F2, in dB

Direct3 = directivity for $F2 <$ frequency \leq F3, in dB

Direct4 = directivity for frequency $>$ F3, in dB

ZRef = reference impedance for all ports

Temp = temperature, in degrees C

CheckPassivity = check passivity flag: if set to yes (default), a passivity check is performed and a warning is output if the device is not passive; if set to no, a passivity check is not performed.

Notes/Equations

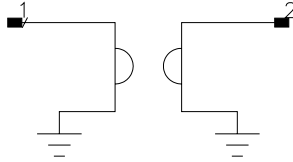
1. Port 1 is the input port; port 2 is the through port; port 3 is the coupled port.
2. Both ports of the main arm (ports 1 and 2) are assumed to have the same VSWR.
3. *Coupling* is assumed to be constant across all frequency bands.
4. *Loss* is the dissipation of the coupler; it affects S21 and S12 only. These S-parameters are calculated as if the coupler was dissipationless and are then scaled by

$$10^{(-abs(Loss)/20)}$$

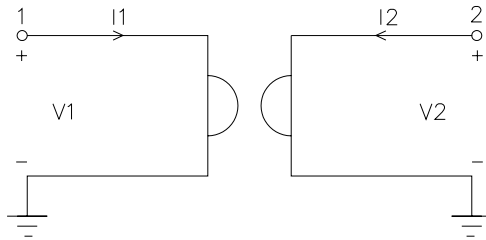
5. Internally, coupler isolation is calculated as *Coupling + Direct* in dB.
6. Up to three frequency breakpoints can be used to define four bands, with different directivity losses and VSWR values for each frequency band. If no frequencies are specified, then this model is frequency independent and uses Direct1, Loss1, and VSWR1 only.
7. There is a 90-degree phase shift between the input and the coupled ports.
8. In the coupled arm, the two ports are uncoupled.

Gyrator (Gyrator)

Symbol



Illustration



Parameters

None

Notes/Equations

$$1. V_1 = \text{Ratio} \times I_2$$

$$V_2 = -\text{Ratio} \times I_1$$

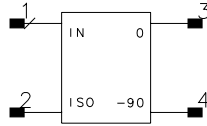
$$S_{11} = S_{22} = \frac{r^2 - 1}{r^2 + 1}$$

$$S_{21} = -S_{12} = \frac{-2r}{r^2 + 1}$$

$$r = \frac{\text{Ratio}}{Z_0}$$

Hybrid90 (Ideal 90-degree Hybrid Coupler)

Symbol



Parameters

Loss = insertion loss, in dB

GainBal = gain balance between output ports, in dB

PhaseBal = phase balance between output ports, in degrees

Rref = port reference impedance, in specified units

Temp = temperature, in °C

CheckPassivity = check passivity flag: if set to yes (default), a passivity check is performed and a warning is output if the device is not passive; if set to no, a passivity check is not performed.

Delay = time delay for output ports

Range of Usage

Loss ≥ 0 dB

Delay $\geq 1.0\text{e-}16$ sec

Notes/Equations

1. Although hybrid couplers are typically designed to give an equal power split between the coupled and the direct ports, the choice of coupling coefficients has been provided here for greater flexibility where a coupling of other than 3 dB is required.
2. The loss term is applied to both the coupling and the direct transmission coefficients.

3. Pin designations (directional coupler notation/0-90 notation):

(IN) = input port #1

(IS0) = isolated output/input port #2

(0) = coupled output / in-phase output

(-90) = direct output/90-degree phase output

$$|S_{0, IN}|(dB) = -10 \times \log(1 + 10^{(-GainBal/10)}) - Loss$$

$$|S_{-90, IN}|(dB) = -10 \times \log(1 + 10^{(-GainBal/10)}) - Loss - GainBal$$

$$|S_{0, ISO}|(dB) = -10 \times \log(1 + 10^{(-GainBal/10)}) - Loss - GainBal$$

$$|S_{-90, ISO}|(dB) = -10 \times \log(1 + 10^{(-GainBal/10)}) - Loss$$

$$phase(S_{0, IN}) = -360 \times Delay \times frequency(degrees)$$

$$phase(S_{-90, IN}) = -360 \times Delay \times frequency - 90 - PhaseBal(degrees)$$

$$phase(S_{0, ISO}) = -360 \times Delay \times frequency - 90 - PhaseBal(degrees)$$

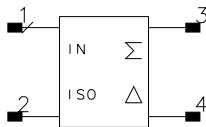
$$phase(S_{-90, ISO}) = -360 \times Delay \times frequency(degrees)$$

4. In general, Hybrid90 is not recommended for time-domain simulation. For the typical case of PhaseBal=0, time domain simulation will not produce proper results for $S_{-90, IN}$, or $S_{0, ISO}$ because 0j in the frequency domain corresponds to a non-causal impulse response in time domain.

5. This component has no default artwork associated with it.

Hybrid180 (Ideal 180-degree Hybrid Coupler)

Symbol



Parameters

Loss = insertion loss, in dB

GainBal = gain balance between output ports, in dB

PhaseBal = phase balance between output ports, in degrees

Rref = port reference impedance, in specified units

Temp = temperature, in °C

CheckPassivity = check passivity flag: if set to yes (default), a passivity check is performed and a warning is output if the device is not passive; if set to no, a passivity check is not performed.

Delay = time delay for output ports

Range of Usage

Loss ≥ 0 dB

Delay $\geq 1.0e-16$ sec

Notes/Equations

1. Although hybrid couplers are typically designed to give an equal power split between the coupled and the direct ports, the choice of coupling coefficient has been provided here for greater flexibility where a coupling of other than 3 dB is required.
2. The loss term is applied to both the coupling and the direct transmission coefficients.

3. Pin designations (directional coupler notation/sum-difference notation):

(IN) = input port/input port#1

(IS0) = isolated output/input port #2

(Σ) = coupled output/summation output

(Δ) = direct output/difference output

$$|S_{\Sigma, IN}|(dB) = -10 \times \log(1 + 10^{(-GainBal/10)}) - Loss - GainBal$$

$$|S_{\Delta, IN}|(dB) = -10 \times \log(1 + 10^{(-GainBal/10)}) - Loss$$

$$|S_{\Sigma, ISO}|(dB) = -10 \times \log(1 + 10^{(-GainBal/10)}) - Loss$$

$$|S_{\Delta, ISO}|(dB) = -10 \times \log(1 + 10^{(-GainBal/10)}) - Loss - GainBal$$

$$phase(S_{\Sigma, IN}) = -360 \times Delay \times frequency - PhaseBal (degrees)$$

$$phase(S_{\Delta, IN}) = -360 \times Delay \times frequency (degrees)$$

$$phase(S_{\Sigma, ISO}) = -360 \times Delay \times frequency (degrees)$$

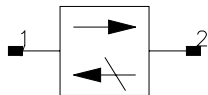
$$phase(S_{\Delta, ISO}) = -360 \times Delay \times frequency - PhaseBal + 180 (degrees)$$

4. When used in time domain simulation, set PhaseBal=0; for any PhaseBal value other than $n \times 180$, time domain simulation will not produce proper results because the frequency domain specification corresponds to a non-causal impulse response in time domain.

5. This component has no default artwork associated with it.

IsolatorSML (SMLIsolator)

Symbol



Parameters

F1 = first frequency breakpoint

F2 = second frequency breakpoint

F3 = third frequency breakpoint

Loss1 = attenuation for frequencies $\leq F1$, in dB

Loss2 = attenuation for frequencies $> F1 \leq F2$, in dB

Loss3 = attenuation for frequencies $> F2 \leq F3$, in dB

Loss4 = attenuation for frequencies $> F3$, in dB

VSWR1 = VSWR at both ports for frequencies $\leq F1$

VSWR2 = VSWR at both ports for frequencies $> F1, \leq F2$

VSWR3 = VSWR at both ports for frequencies $> F2, \leq F3$

VSWR4 = VSWR at both ports for frequencies $> F3$

Isolat = isolation, in dB

Z1 = reference impedance for port 1

Z2 = reference impedance for port 2

Temp = temperature, in degrees C

CheckPassivity = check passivity flag: if set to yes (default), a passivity check is performed and a warning is output if the device is not passive; if set to no, a passivity check is not performed.

Notes/Equations

1. All ports are assumed to have the same VSWR.
2. Isolation is assumed to be constant across all frequency bands.

3. Up to three frequency breakpoints can be used to define four bands, with different losses and VSWR for each frequency band. If frequencies are not specified, this model is frequency independent and uses Loss1 and VSWR1 only.

LOS_Link (Line-Of-Sight Antenna Link)

Symbol



Parameters

CenterFreq = link center frequency, in hertz

BW = link bandwidth, in hertz

TxGain = transmitting antenna gain, in dB

TxVSWR = transmitting antenna VSWR

TxParabolaD = transmitting parabolic antenna diameter, in specified units

TxEfficiency = transmitting parabolic antenna efficiency

RxGain = receiving antenna gain, in dB

RxVSWR = receiving antenna VSWR

RxParabolaD = receiving parabolic antenna diameter, in specified units

RxEfficiency = receiving parabolic antenna efficiency

RxNoiseTemp = receiving antenna noise temperature, in Kelvin

PathLength = line-of-sight path length, in distance units

NotchFreq = notch frequency due to ground reflection path interference, in hertz

NotchDepth = power of reflected ray relative to direct ray, in dB

DeltaDelay = time delay of ground reflection path w.r.t. LOS path, in seconds

Z1 = transmitting antenna reference impedance, in ohms

Z2 = receiving antenna reference impedance, in ohms

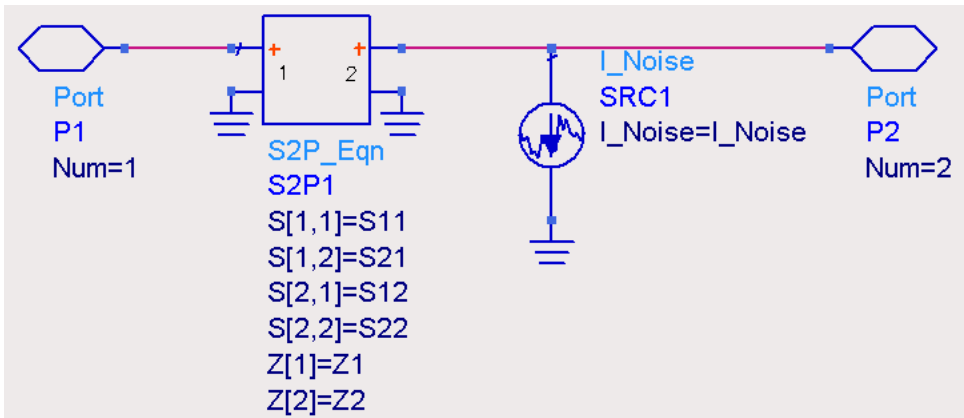
Range of Usage

PathLength > 10 wavelengths

Notes/Equations

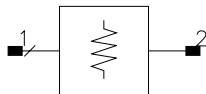
1. Pathloss is infinite outside a window defined by BW.

2. Transmitting antenna gain can be defined by TxGain or by TxParabolaD and TxEfficiency. If all three parameters are specified, TxParabolaD and TxEfficiency will overwrite TxGain. The same applies to receiving antenna as well.
3. The S-Parameter and noise implementation of this model is as follows. S11, S22, S12, S21, and I_Noise are derived from the user-specified LOS_link parameters.



Pad (Pi or Tee Format)

Symbol



Parameters

NetType = network type: Pi or Tee

Loss = attenuation, in dB

R1 = reference resistance for port 1

R2 = reference resistance for port 2

Temp = temperature, in degrees C

CheckPassivity = check passivity flag: if set to yes (default), a passivity check is performed and a warning is output if the device is not passive; if set to no, a passivity check is not performed.

Range of Usage

$R1 > 0$

$R2 > 0$

Notes/Equations

1. This 2-port component provides a:

- Match to resistance R_1 at port 1 when port 2 is terminated in R_2
- Match to R_2 at port 2 when port 1 is terminated in R_1
- Desired attenuation, DB

2. Resistive networks can achieve simultaneous match at input and output if the insertion loss factor A exceeds a minimum value given by

$$A_{min} = 2 \times \frac{R_1}{R_2} \times \left(1 + \sqrt{1 - \frac{R_2}{R_1}} \right) - 1.0 \quad R_1 > R_2$$

$$2 \times \frac{R_2}{R_1} \times \left(1 + \sqrt{1 - \frac{R_1}{R_2}} \right) - 1.0 \quad R_2 > R_1$$

where

$$DB_{MIN} = 10 \times \log_{10} (A_{MIN})$$

3. The values shown in the Pi network equivalent circuit are given by:

$$G_c = 2 \times \frac{\sqrt{A}}{(A-1) \times \sqrt{R_1 \times R_2}}$$

$$G_a = \frac{A+1}{(A-1) \times Z_1} - G_c$$

$$G_b = \frac{A+1}{(A-1) \times R_2} - G_c$$

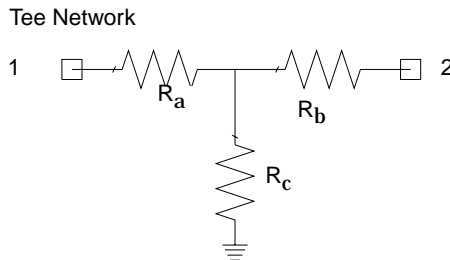
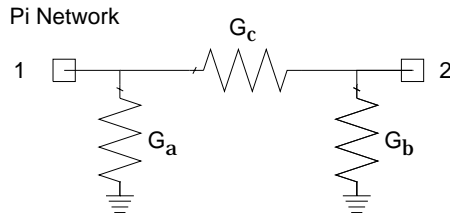
The values shown in the Tee network circuits are given by

$$R_c = \frac{2 \times \sqrt{A} \times \sqrt{R_1 \times R_2}}{(A-1)}$$

$$R_a = \frac{R_1 \times (A+1)}{A-1} - R_c$$

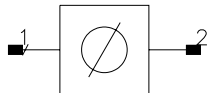
$$R_b = \frac{R_2 \times (A+1)}{A-1} - R_c$$

Equivalent Circuit



PhaseShiftSML (Phase Shifter)

Symbol



Parameters

Phase = constant phase shift, in specified units

PhaseSlope = phase slope per frequency octave

FreqStart = frequency where slope begins

RTConj = reverse transmission conjugate: NO, YES

ZRef = reference impedance for all ports

Range of Usage

FreqStart ≥ 0

Notes/Equations

1. The output frequency spectrum is equal to the input spectrum shifted by the specified phase.

2. $\theta(f) = \text{Phase}$ (for $\text{freq} < \text{FreqStart}$)

$\theta(f) = \text{Phase} + \text{PhaseSlope} \times \text{LOG}_2(\text{freq} / \text{FreqStart})$ (for $\text{freq} \geq \text{FreqStart}$)

if $\text{FreqStart} = 0$, then phase slope is 0 regardless of the PhaseSlope setting
where freq = simulation frequency

$$3. \frac{V_2}{V_1} = \frac{I_2}{I_1} = e^{j\theta(f)}$$

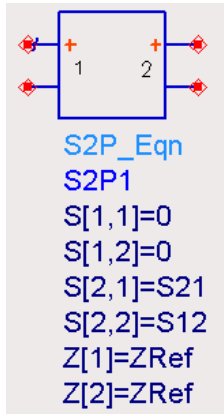
$$S_{11} = S_{22} = 0$$

$$S_{21} = e^{j\theta(f)}$$

if RTConj = NO, $S_{12} = S_{21}$

if RTConj = YES, $S_{12} = S_{21}^*$

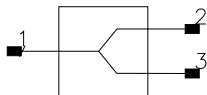
4. The S-parameter implementation for the phase shifter is shown here:



5. In general, PhaseShiftSML is not recommended for time domain simulation. For the special case of $n \times 180^\circ$ phase shift, time domain simulation produces proper results; for the extreme case of $n \times 90^\circ$ phase shift, time domain simulation will not produce proper results because $(\pm)j$ in the frequency domain corresponds to a non-causal impulse response in time domain.

PwrSplit2 (2-Way Power Splitter)

Symbol



Parameters

S_{21} = port 1 to port 2 complex transmission coefficient

S_{31} = port 1 to port 3 complex transmission coefficient

S_{11} = port 1 complex reflection coefficient

S_{22} = port 2 complex reflection coefficient

Isolation = isolation between port 2 and port 3, in dB

ZRef = reference impedance for all ports

Temp = temperature, in degrees C

CheckPassivity = check passivity flag: if set to yes (default), a passivity check is performed and a warning is output if the device is not passive; if set to no, a passivity check is not performed.

Delay = time delay, in seconds

Notes/Equations

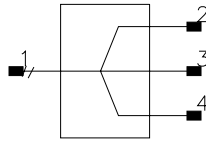
1. $S_{12} = S_{21}$, $S_{13} = S_{31}$

2. PwrSplit2 can also be used as a power combiner.

3. Use the functions polar(), dbpolar() to represent the S-parameters in terms of magnitude, phase or dB, phase.

PwrSplit3 (3-Way Power Splitter)

Symbol



Parameters

S21 = port 1 to port 2 complex transmission coefficient

S31 = port 1 to port 3 complex transmission coefficient

S41 = port 1 to port 4 complex transmission coefficient

ZRef = reference impedance for all ports

Temp = temperature, in degrees C

CheckPassivity = check passivity flag: if set to yes (default), a passivity check is performed and a warning is output if the device is not passive; if set to no, a passivity check is not performed.

Delay = time delay, in seconds

Notes/Equations

1. Ideal isolation exists for S_{23} , S_{24} , S_{32} , S_{34} , S_{42} , and S_{43} , that is,

$$S_{23} = S_{32} = S_{24} = S_{42} = S_{34} = S_{43} = 0.$$

2. Ideal match exists for S_{11} , S_{22} , S_{33} , and S_{44} , that is,

$$S_{11} = S_{22} = S_{33} = S_{44} = 0.$$

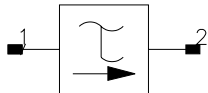
3. $S_{12} = S_{21}$, $S_{13} = S_{31}$ and $S_{14} = S_{41}$.

4. PwrSplit3 can also be used as a power combiner.

5. Use the functions `polar()`, `dBpolar()` to represent the S-parameters in terms of magnitude, phase or dB, phase.

TimeDelay (Time Delay)

Symbol



Parameters

Delay = time delay, in seconds (default = 1 μ sec)

RTConj = reverse transmission conjugate: NO (default), YES

ZRef = reference impedance for all ports, in ohms (default=50. Ohm)

Range of Usage

Delay ≥ 0

Notes/Equations

1. The input frequency spectrum has a linear phase shift applied resulting in the output time waveform being a time shifted replication of the input waveform.

2. $S_{21} = e^{-j\omega\text{Delay}}$

$$S_{11} = S_{22} = 0$$

$$\omega = 2\pi f$$

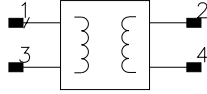
if RTConj = NO, $S_{12} = S_{21}$

if RTConj = YES, $S_{12} = S_{21}^*$

3. Excessive values for the Delay parameter require large memory for Circuit Envelope and Transient simulations. For Circuit Envelope simulations, a safeguard limits any excessive delay to one-half of the stop time.

Transformer (Ideal 4-Port Transformer)

Symbol



Parameters

None

Range of Usage

$N \neq 0$

Notes/Equations

Note This component is obsolete for new designs. (It is available only for compatibility with designs created with 2002C or earlier releases.)

Please use the TF transformer (Lumped-Components library) for new design work.

1. Input pins are 1 and 3; output pins are 2 and 4.
2. The S-parameters of the component are determined as follows:

$$S_{11} = S_{24} = S_{33} = S_{42} = N^2 / (1 + N^2)$$

$$S_{14} = S_{23} = S_{32} = S_{41} = -N / (1 + N^2)$$

$$S_{12} = S_{21} = S_{34} = S_{43} = N / (1 + N^2)$$

$$S_{13} = S_{22} = S_{31} = S_{44} = 1 / (1 + N^2)$$

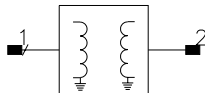
3. The input resistance R_{IN} output resistance R_{OUT} are related to N by:

$$\sqrt{\frac{R_{IN}}{R_{OUT}}} = N$$

4. This is a noiseless component.
5. Because it is an ideal transformer, the impedance transformation is the same at DC as it is at nonzero frequencies.
6. This component passes DC.

TransformerG (Transformer with Ground Reference)

Symbol



Parameters

None

Range of Usage

$N \neq 0$

Notes/Equations

Note This component is obsolete for new designs. (It is available only for compatibility with designs created with 2002C or earlier releases.)

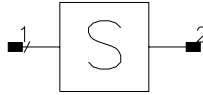
Please use the TF transformer (Lumped-Components library) for new design work.

1. The input resistance R_{IN} and R_{OUT} are related to N by:

$$\sqrt{\frac{R_{IN}}{R_{OUT}}} = N$$

TwoPort (2-Port Model)

Symbol



Parameters

S21 = complex forward transmission coefficient

S12 = complex reverse transmission coefficient

S11 = port 1 complex reflection coefficient

S22 = port 2 complex reflection coefficient

ZRef = reference impedance for all ports

Temp = temperature, in degrees C

CheckPassivity = check passivity flag: if set to yes (default), a passivity check is performed and a warning is output if the device is not passive; if set to no, a passivity check is not performed.

Delay = time delay, in seconds

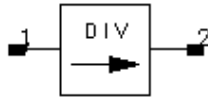
Notes/Equations

1. Use the functions `polar()`, `dBpolar()` to represent the S-parameters in terms of magnitude, phase or dB, phase.

Chapter 5: Phase Lock Loop Components

DivideByN (Divide by N)

Symbol



Parameters

FnomIn = nominal input frequency, in hertz

N = divide number

Notes/Equations

1. This model performs a divide-by-N function on either a baseband-input or a selected-carrier input frequency. The model operates in transient, harmonic balance, or circuit envelope simulation.

In transient, all signals are considered baseband.

In circuit envelope, the FnomIn parameter defines which analysis frequency to use. If the analysis frequency is not within $0.5/\text{timestep}$ a warning is issued and 0 Hz will be used for the analysis frequency.

2. The input impedance is infinite. The output impedance fixed at 1 ohm. If the output is in baseband mode, then its open circuit voltage is equal to the phase of the divided signal in radians. This results in a sawtooth waveform for a CW input. A sawtooth is output (instead of a square wave) to allow for the calculation of the actual frequency from the slope and for the accurate detection of zero crossings without the sampling jitter that would be introduced by the fixed rate sampling of a square wave. If the output is not in baseband mode, then the open circuit voltage is a 1V complex sinusoid at the divided frequency. Whether or not the output is in baseband is determined, in Circuit Envelope mode, by determining the carrier frequency, possibly dc, that is closest to FnomIn; if none are close enough, a warning is issued.
3. In non-baseband mode of operation, the divider works by directly extracting the phase of the complex input. The delay of DivideByN = $\max(\text{timestep}, N/(2 \times F_{in}))$, where F_{in} is the actual analysis frequency corresponding to FnomIn.

In baseband mode of operation, the period of the input is determined by detecting when the baseband signal rises through the 0.5V threshold. From this

period information, the phase of the divided signal is calculated. For a more standard counter model, see the counter model under probes.

4. **Figure 5-1** show examples of the divider operating with an RF carrier input: one divider outputs a baseband output; one divider outputs the divided frequency at the other carrier frequency. Note that N can be time varying in order to simulate the effect of a fractional-N divider. **Figure 5-2** shows the simulation results.

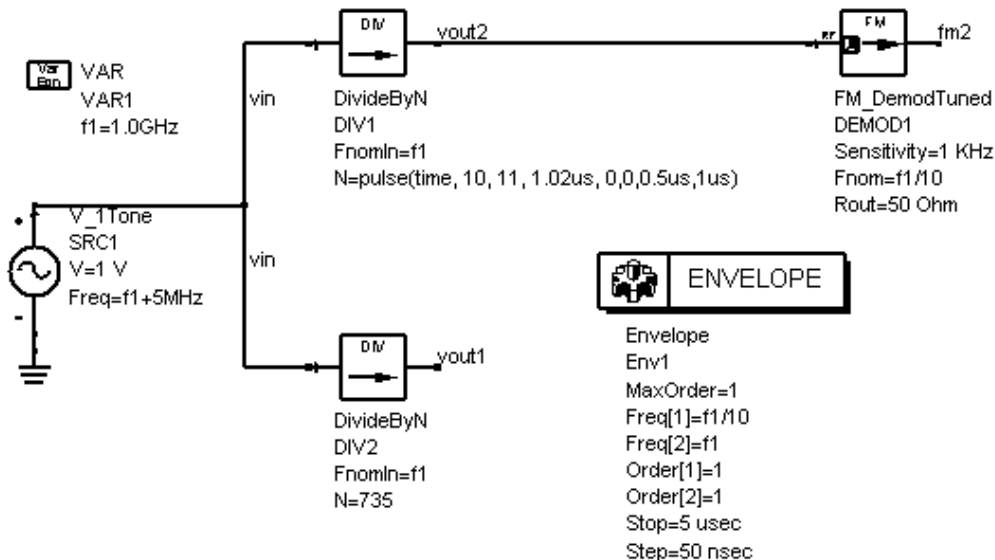


Figure 5-1. Divider with an RF Carrier Input

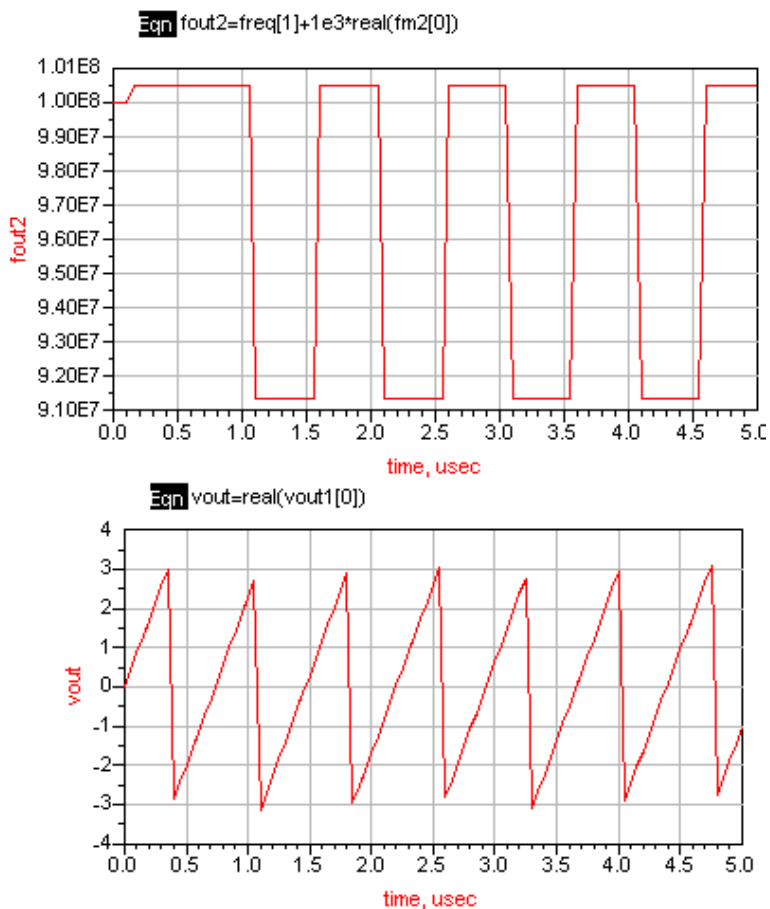
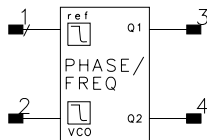


Figure 5-2. Output Waveforms

PhaseFreqDet (Frequency Detector, Baseband)

Symbol



Parameters

Vhigh = High-state output voltage

Vlow = Low-state output voltage

Notes/Equations

1. PhaseFreqDet is not selectable from the component palette or component library browser; to place this component, type its exact name into the Component History box above the drawing area, then move your cursor to the drawing area.
2. This baseband phase-frequency demodulator is used in transient or circuit envelope simulation. It models the digital behavior of common D flip-flop type phase-frequency detectors often used in phase-locked loops. The two outputs are ideal, zero impedance voltage sources; to model a pulsed current-source output, two VDCS-dependent sources must be added to the output of this model. The two inputs have infinite impedance, and only the baseband portion of the two input voltages are used to determine threshold-crossing timing.
3. As opposed to the tuned phase-frequency detector model, this model's output includes the effect of reference clock feed-through.

Note The output of this model is a pulse train whose average value is proportional to the input phase difference, and may contain significant signal energy at the reference clock rate and at clock harmonics. These must be filtered out, typically before driving a VCO in a PLL application. The tuned phase-frequency detector output signal includes the instantaneous phase difference information only; it does not contain reference frequency or harmonic content.

However, the penalty for this is that the timestep must be less than one-half the reference period, and typically less than one-tenth the period. To avoid the large

amount of time jitter and phase noise that would normally be introduced by sampling at even these rates, the two digital outputs are also amplitude modulated to reflect the portion of a simulation timestep where the actual outputs would be high or low. For example, if based on the threshold-crossing timing (the pulse width should be 10 nsec, but the simulation timestep is 1 μ sec), then the output amplitude for that timestep would only be 1% of the V_{high} level. While this will not properly model all the higher harmonics of the reference feed-through, it does accurately model both the dc term and the first few harmonics, and the corresponding reference sidebands of the VCO. As the analysis timestep is further reduced, the behavior becomes more truly digital in nature and more harmonics are effectively being simulated at the cost of slower simulations.

4. The trigger times for both inputs is determined by detecting when the baseband voltage rises through the 0.5V threshold. Linear interpolation is used to get much finer time resolution than the analysis timestep. To further reduce excessive sampling jitter, the inputs to this detector can be sawtooth waveforms. While the detector will work reasonably well with sinusoidal inputs, given a small enough timestep, timing jitter can be eliminated if the interpolation is done on the positive slope of a sawtooth waveform. This is the reason why the divide-by-N models output a sawtooth waveform when they operate in the baseband mode. Square-wave inputs should generally be avoided, because this will usually introduce significant timing jitter and phase noise into the simulation.
5. This model does not include any effects due to the finite duration of the flip-flop reset pulse and resultant zero-phase dead zone. Other effects, such as asymmetry between the two different outputs, can be incorporated by changing the external components. [Figure 5-3](#) shows an example using this detector; [Figure 5-4](#) shows the output waveforms.

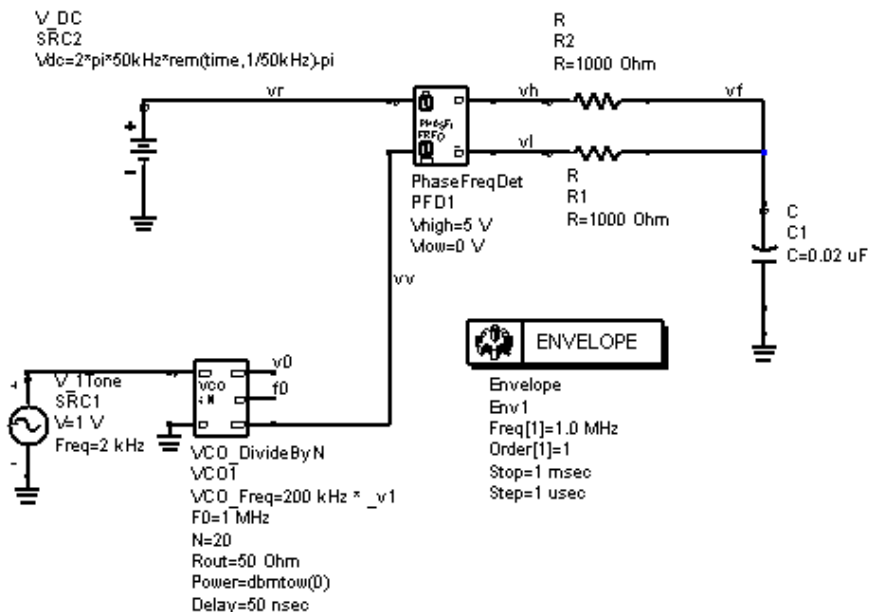


Figure 5-3. PhaseFreqDet Example

$$\text{Eqn } v_{out} = \text{real}(v_f[0])$$

$$\text{Eqn } v_{high} = \text{real}(v_h[0])$$

$$\text{Eqn } v_{low} = \text{real}(v_l[0])$$

$$\text{Eqn } v_{ref} = \text{real}(v_r[0])$$

$$\text{Eqn } v_{vcon} = \text{real}(w[0])$$

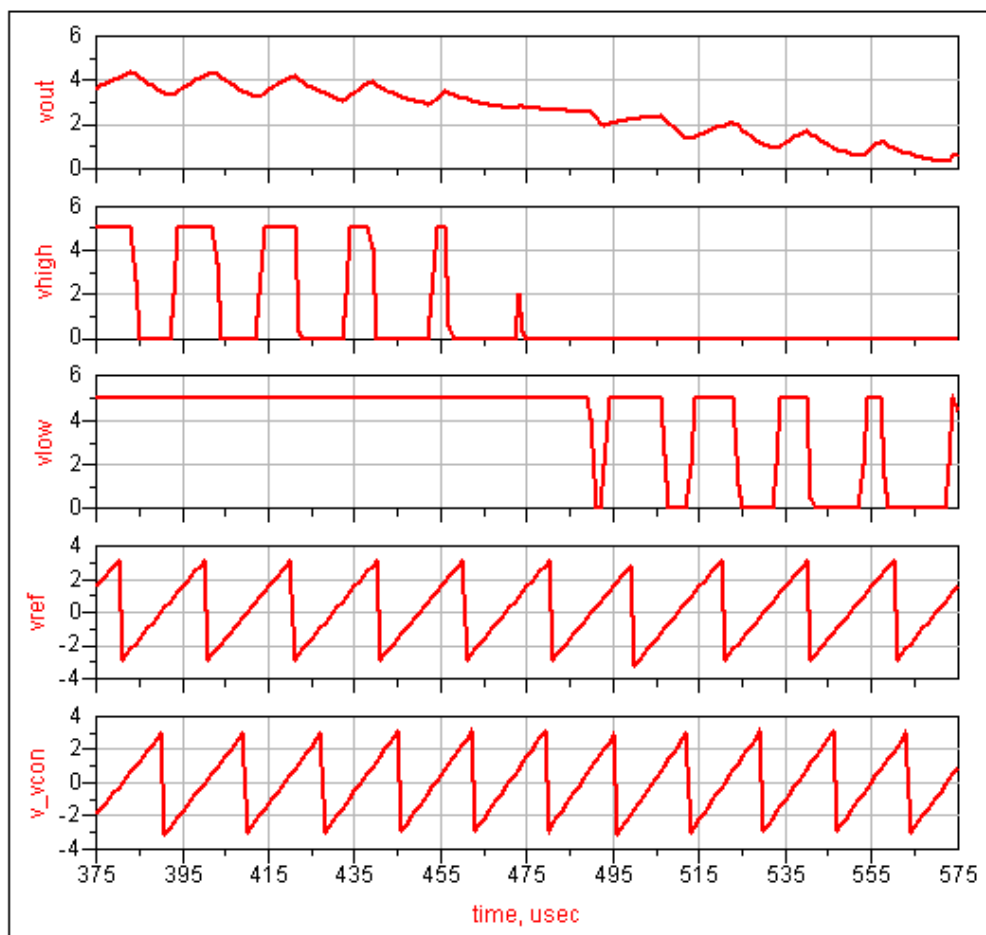
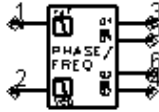


Figure 5-4. Output Waveforms

PhaseFreqDet2 (Frequency Detector, Baseband)

Symbol



Parameters

Vhigh = high-state output voltage

Vlow = low-state output voltage

DeadTime = dead zone pulse width

Jitter = input time jitter

Notes/Equations

1. This baseband phase-frequency demodulator is used in transient or circuit envelope simulation. It models the digital behavior of common D flip-flop type phase-frequency detectors often used in phase-locked loops. The four outputs are ideal, zero impedance voltage sources. The two inputs have infinite impedance, and only the baseband portion of the two input voltages are used to determine threshold-crossing timing.
2. The following FDD modeling equations illustrate how the shape of the four outputs relate to the input phase difference.

The phase difference between the input on port 1 and port 2 is calculated as

$ns = _phase_freq(1,2)$, and the quantities $n1 = (V_{high} - V_{low}) \times \text{real}(ns)$ and $n2 = (V_{high} - V_{low}) \times \text{imag}(ns)$ are introduced.

Outputs on ports 3-6 are then

port 3: $Q1 = V_{low} + n1$

port 4: $Q2_bar = V_{high} - n2$

port 5: $Q1_bar = V_{high} - n1$ (only PhaseFreqDet2, not PhaseFreqDet)

port 6: $Q2 = V_{low} + n2$ (only PhaseFreqDet2, not PhaseFreqDet)

with the following trigger events:

Trig[1] = $_xcross(1,0.5,1)$

Trig[2] = $_xcross(3,0.5,1)$

For details about these functions, refer to the FDD device documentation in the *Circuit Components Nonlinear Devices* manual.

Note that you can push into the component to see the implementation of the component.

3. As opposed to the tuned phase-frequency detector model, this model's output includes reference clock feed-through effects.

Note The output of this model is a pulse train whose average value is proportional to the input phase difference, and may contain significant signal energy at the reference clock rate, and at clock harmonics. Typically, these must be filtered out before driving a VCO in a PLL application. The tuned phase-frequency detector output signal includes instantaneous phase difference information only (it does not contain reference frequency or harmonic content).

However, the penalty for this is that the timestep must be less than one-half the reference period, and typically less than one-tenth the period. To avoid the large amount of time jitter and phase noise that would normally be introduced by sampling at even these rates, the four digital outputs are also amplitude modulated to reflect the portion of a simulation timestep where the actual outputs would be high or low. For example, if based on the threshold-crossing timing (the pulse width should be 10 nsec, but the simulation timestep is 1 μ sec), then the output amplitude for that timestep would only be 1% of the Vhigh level. While this will not properly model all the higher harmonics of the reference feed-through, it does accurately model both the dc term and the first few harmonics, and the corresponding reference sidebands of the VCO. As the analysis timestep is further reduced, the behavior becomes more truly digital in nature and more harmonics are effectively being simulated at the cost of slower simulations.

4. The trigger times for both inputs is determined by detecting when the baseband voltage rises through the 0.5V threshold. Linear interpolation is used to get much finer time resolution than the analysis timestep. To further reduce excessive sampling jitter, inputs to this detector can be sawtooth waveforms. While the detector will work reasonably well with sinusoidal inputs, given a small enough timestep, the timing jitter can be eliminated if the interpolation is done on the positive slope of a sawtooth waveform. This is the reason why the divide-by-N models output a sawtooth waveform when they operate in the baseband mode. Square wave-inputs should generally be avoided, because this

will usually introduce significant timing jitter and phase noise into the simulation.

5. DeadTime specifies the period of time centered around the 0 phase output during which no output is generated from either the high or the low charge pump. Outside of this period, the output returns to the ideal pulse widths and amplitudes determined by the trigger crossings of the inputs and the charge pump currents.
6. The Jitter value defines the RMS time jitter associated with a trigger crossing on either input. The distribution is Gaussian and the noise spectrum is assumed flat out to the reference sampling frequency.
7. Other effects, such as asymmetry between the four different outputs, can be incorporated by changing the external components. [Figure 5-5](#) shows one application of this detector and [Figure 5-6](#) the resultant output waveforms.

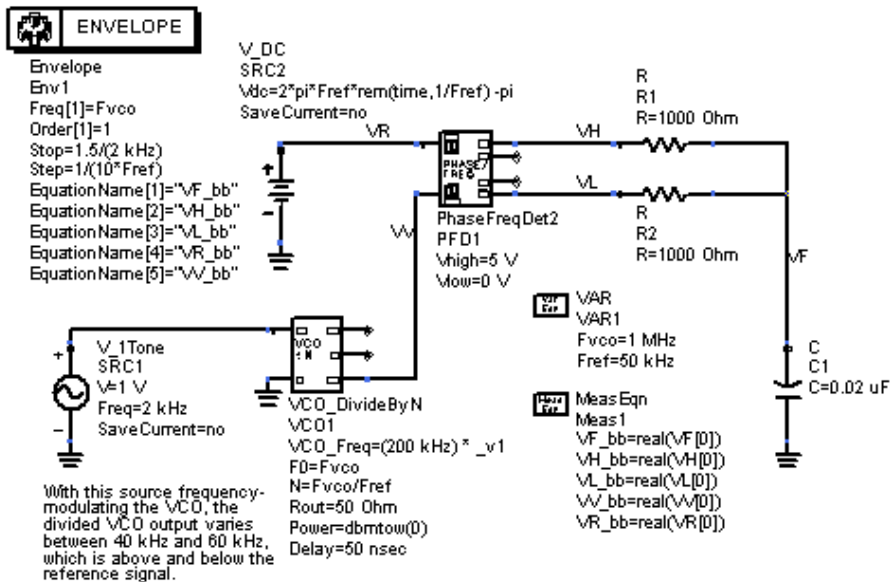


Figure 5-5. One application of detector

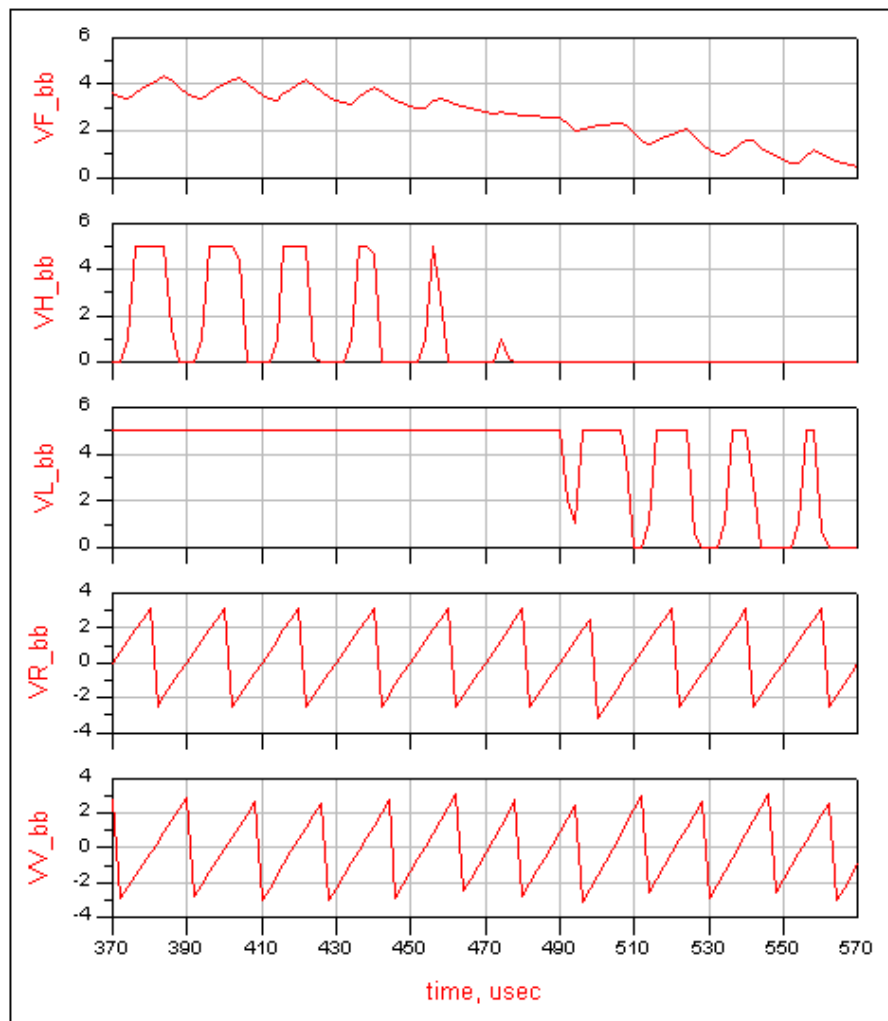
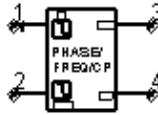


Figure 5-6. Resultant output waveforms of detector

PhaseFreqDetCP (Frequency Detector, Baseband with Charge Pump)

Symbol



Parameters

Ihigh = high-level charge pump current

Ilow = low-level charge pump current

DeadTime = dead-zone pulse width

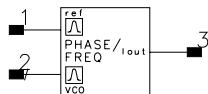
Jitter = input time jitter

Notes/Equations

1. This baseband phase-frequency demodulator with charge pump is used in transient or circuit envelope simulation.
2. Charge pump currents can be constants (in which case they are ideal current sources) or functions of the state variable `_v2` to allow for a non-ideal current whose peak value varies as a function of the output voltages. In either case, the off-state current is always 0.0. Normally positive values would be used for both currents, as Ihigh specifies a source current and Ilow specifies the sink current.
3. DeadTime specifies the period of time centered around the 0 phase output during which no output is generated from either the high or the low charge pump. Outside of this period, the output returns to the ideal pulse widths and amplitudes determined by the trigger crossings of the inputs and the charge pump currents.
4. The Jitter value defines the RMS time jitter associated with the trigger crossing at each of the two inputs. The effective output time jitter will be $\sqrt{2}$ times this value. The distribution is Gaussian and the noise spectrum is assumed flat out to the reference sampling frequency.

PhaseFreqDetTuned (Phase Frequency Detector, Tuned)

Symbol



Parameters

Sensitivity = detector sensitivity, in mA/degree

MaxAngle = maximum unwrapped phase angle, in degrees

Vlimit = maximum output voltage compliance, in volts

Fnom = nominal input frequency for VCO and REF inputs, in hertz

Notes/Equations

1. This tuned phase-frequency demodulator is used with circuit envelope simulation that models the ideal behavior of the phase frequency detectors used in phase-locked loops. (This model does not work with transient simulation.)

On the REF and the VCO inputs, it selects the input carrier closest to the specified Fnom frequency. For proper operation, this cannot be the baseband (dc) envelope. It then generates a baseband output signal equal to the phase difference between the VCO and REF inputs, with an offset of $2 \times \pi \times \Delta f \times \text{time}$ being generated if there is a frequency difference Δf between the two inputs.

As an example consider the sources $V1(t) = \exp(j \times P1(t))$ at $f0$ and $V2(t) = \exp(j \times P2(t))$ also at $f0$, as input to a PhaseFreqDetTuned component with the output terminated in a high impedance. Envelope simulation at $f0$ yields a DC output with a $P1(t) - P2(t)$ time dependence. Changing the frequency of the second source from $f0$ to $f0 + \Delta f$ for the same $V2(t)$ is equivalent with keeping the frequency at $f0$ but multiplying $V2(t)$ by $\exp(j \times 2 \times \pi \times \Delta f \times t)$ and results in the output signal $P1(t) - P2(t) - (2 \times \pi \times \Delta f \times t)$.

If there is no analysis harmonic frequency close enough to the Fnom frequency, then a warning is generated. The two input impedances are infinite. The output is a baseband-only infinite impedance current source with scaling defined by the Sensitivity parameter. If the output voltage exceeds Vlimit or is lower than $-Vlimit$, then a voltage limiter with an impedance of 10 ohms is switched in.

The MaxAngle parameter can be used to model special phase-frequency detectors that have greater than the traditional $\pm 2\pi$ of input phase range.

2. This model does not model any of the reference or VCO frequency components on the output. The output is a baseband-only signal, and the voltage limiting applies to the baseband component only. (See the baseband phase-frequency detector for a model that can overcome this limitation.) With this tuned model, though, the timesteps can actually be larger than the reference period, and it is only dependent on the PLL bandwidth and not the reference frequency; this enables faster performance for many types of simulation measurements.

Another limitation is that only the phase of the selected input envelope is included. The effect of additional input spurs and harmonics are not included if they are in different carrier frequency envelopes.

3. [Figure 5-7](#) and [Figure 5-8](#) show a simple schematic and example of this phase-frequency detector's output. The reference input is a fixed frequency. The VCO input has sinusoidal frequency modulation.

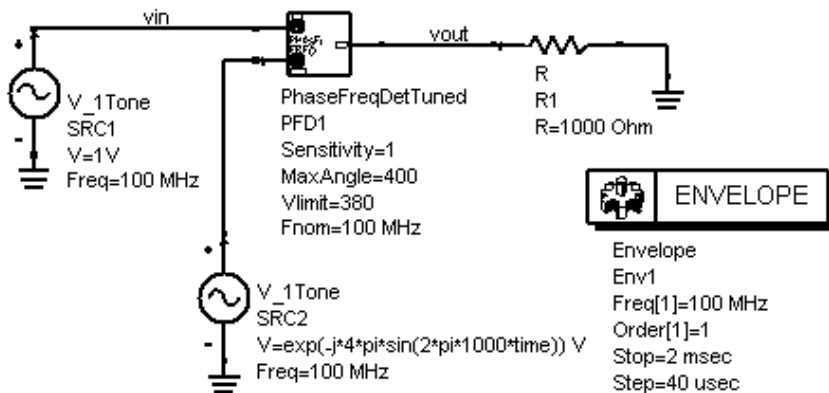


Figure 5-7. Phase-Frequency Detector Example

Eqn $V_{out1} = \text{real}(v_{out}[0])$

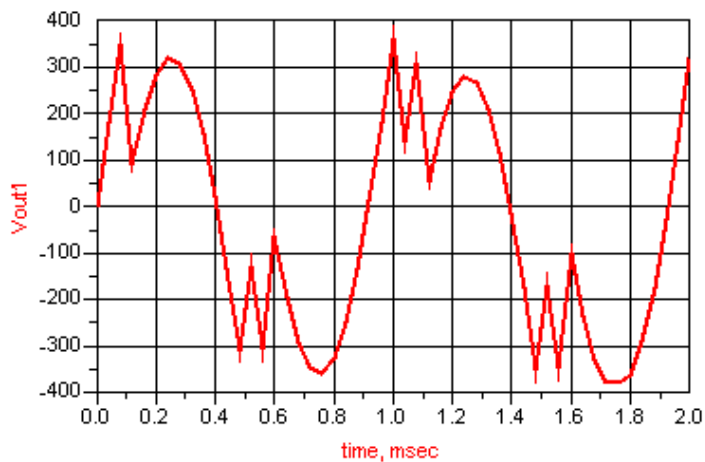
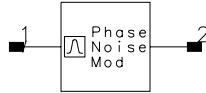


Figure 5-8.

PhaseNoiseMod (Phase Noise Modulator)

Symbol



Parameters

Fnom = nominal input frequency, in hertz

Rout = output resistance, in ohms

Fcorner = corner frequency for 1/f noise performance, in hertz

NF = broadband noise figure, in dB

QL = loaded Q of resonator

Notes/Equations

1. This device uses Leeson's equation to model oscillator phase noise, then modulates the input carrier with this phase noise.

The input can be from any signal source, including the VCO models. This model behaves as a tuned modulator by selecting and modulating just the carrier defined by the Fnom parameter. If there are no analysis frequencies close enough to this value, a warning is generated and no signal is output. The Leeson's equation models the oscillator phase using the equation

$$S_{\Delta\phi} = \frac{(F)kTB}{P_{sig_{AV}}} \times \left(1 + \left(\frac{F_{nom}}{2Q_L Freq}\right)^2\right) \times \left(1 + \frac{F_{corner}}{Freq}\right)$$

where

Fnom is used as an approximation to carrier frequency

F is the noise factor

$P_{sig_{AV}}$ is the input signal power

T is the absolute temperature

B is the analysis bandwidth

k is the Boltzmann constant

Q_L is the loaded Q value of the oscillator's resonator

Fcorner is the frequency at which the low frequency 1/f noise is equal to the broadband noise.

This model is usable in frequency domain and circuit envelope time domain noise analyses. To avoid the divide-by-zero problems as the analysis offset frequency approaches 0, both the $1/f^2$ and $1/f$ terms are rolled off at frequencies below 1 Hz. In the time-domain mode, the $1/f$ frequency response is implemented by doing a convolution simulation. The duration of this impulse response is set to 2000 timesteps. This effectively rolls off this $1/f$ response at a frequency determined by the analysis tstep parameter.

VCO (Voltage Controlled Oscillator)

Symbol



Parameters

K_v = frequency tuning sensitivity, in hertz per volt

Freq = fundamental frequency, in hertz

P = power into R_{out} load at fundamental frequency

R_{out} = output resistance, in ohms

Delay = transit time delay added to input tuning voltage, in seconds

Harmonics = ratio of harmonic voltage to fundamental voltage, complex value

Notes/Equations

1. This is a simple VCO model that outputs a signal whose frequency is controlled in a linear manner by the input tuning voltage. The center frequency is defined by the Freq parameter, which references one of the analysis frequencies. A Freq value of 0 references dc and so defines the VCO as a baseband source; the output frequency is the baseband input voltage times the K_v parameter value. Only the baseband portion of the input tuning voltage is used to determine the VCO frequency offset.
2. The phase of the VCO output is clamped when time equals 0, so this model only functions as a VCO in the time-domain analysis modes, including circuit envelope and transient simulation. Because transient simulation is a baseband-only analysis mode, Freq should be 0 in this mode.
3. The output resistance at all frequencies is set by R_{out} , which is internally limited to a minimum value of 0.1 ohm. The VCO's fundamental output power into an R_{out} load is defined by the P parameter. The relative level, in linear units, of the second and third harmonics are defined by the Harmonics parameter; this parameter can be complex to allow definition of the relative phase of these harmonics. The initial phase for the fundamental frequency output is set to -90 degrees so that a sine waveform is created in baseband mode.

4. A Delay parameter value can also be specified for this VCO model. This puts an additional transit delay between the input tuning voltage and the actual change in the output frequency. A delay of at least one time-step does sometimes result in faster simulation speeds, and can be used to model the time delay inherent in any real VCO.

Care should be taken when using the VCO in baseband mode (Freq = 0). If the input voltage is allowed to go negative, the model will generate a negative frequency, and the waveform shape due to the harmonic content will be reversed.

The VCO_DivideByN model also allows the definition of a nonlinear frequency tuning characteristic.

5. **Figure 5-9** shows the use of the VCO in a transient simulation. The VCO is defined with significant second and third harmonic levels. Simulation results are shown in **Figure 5-10**.

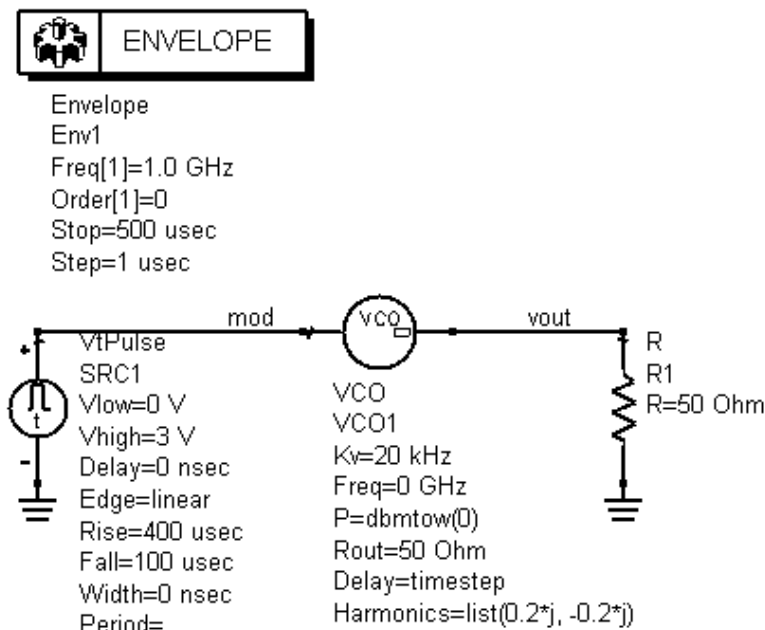


Figure 5-9. VCO in a transient simulation

`Eqn mod_out=real(mod[0])`

`Eqn vout_out=real(vout[0])`

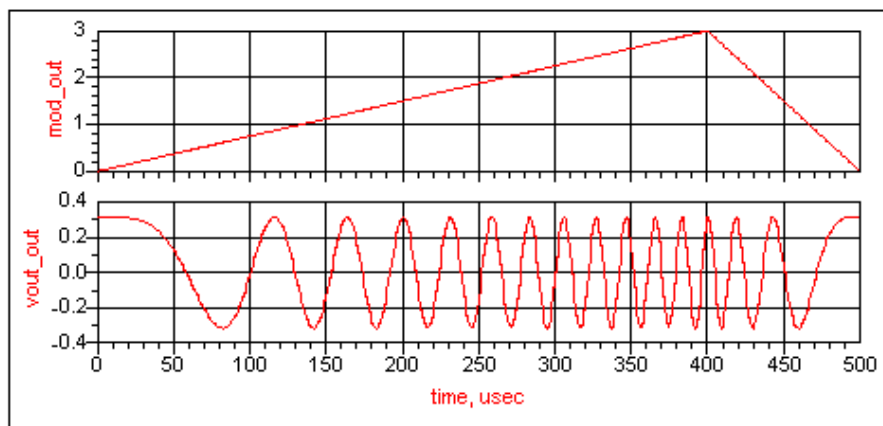
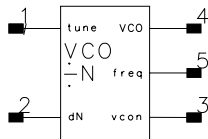


Figure 5-10. Simulation Results

VCO_DivideByN (VCO Divide By N)

Symbol



Parameters

VCO_Freq = frequency deviation from F0 (function of v_1), in hertz per volt

F0 = VCO center frequency, in hertz

N = nominal divide number (with $dN=0$)

Rout = output resistance of VCO, in ohms

Power = output power into Rout load, in watts

Delay = transit time delay added to input tuning voltage, in seconds

Pin Connections

tune connects to a tuning voltage.

dN connects to ground ($dN=0$) or a voltage source such as V_{DC} .

dN takes the value of nodal voltage and $N+dN$ becomes the divide ratio.

vcon is the divided-by- $(N+dN)$ output.

freq outputs the undivided frequency values from pin VCO. This pin can be left open.

VCO is the undivided VCO output. This pin can be left open if an undivided VCO is not used.

Notes/Equations

1. This VCO model allows for the definition of an arbitrary, nonlinear frequency tuning characteristic. In addition, it incorporates a behavioral, divide by N model. Incorporating the divider into the same model permits its use in phase-lock loop simulations where the envelope bandwidth, as determined by the analysis time-step, does not have to include the entire tuning range of the VCO, but only the frequency range of the divided output. In these cases, the phase and frequency information of the VCO's main output may be aliased because of the large time-step. But, if just the divided output is being used, the

loop simulations will still be valid and can simulate faster due to the large time-step.

As the time-step is decreased, or the range of the VCO is reduced, such that it remains within the envelope bandwidth, then both the main VCO and the divided VCO output are valid. As with the standard VCO model, the phase of this model's outputs are also clamped when time=0 so this model only functions as a VCO in the time-domain analysis modes, including circuit envelope and transient simulations.

2. The frequency of the VCO is determined by the F0 value plus the VCO_Freq value. The VCO_Freq value may be an arbitrary expression using `_v1`, which is a pre-defined variable representing the input tuning voltage. The frequency is determined inside the model by determining the present value of `_v1` (the input voltage), evaluating the VCO_Freq expression and adding this to the F0 value. This frequency is output, as an ideal voltage source scaled to 1V per GHz, on the freq output pin.
3. In circuit envelope simulation, the carrier frequency envelope associated with the main VCO frequency is determined by the F0 value. If there is no analysis harmonic frequency close enough to F0, then a warning is generated and the main output is zero. The carrier frequency envelope associated with the divided VCO frequency is determined by $F0/N$. Again, if no analysis harmonic frequency is close enough, then a warning is generated and this output is also set to zero.
4. The divide number is determined by adding the N parameter and the dN baseband input voltage. The divide number can change during the simulation. By properly driving the dN input, fractional frequency division can be simulated. To simulate all the dynamics of a fractional divider, the simulation time-step must be small enough to properly digitize the varying divide- or pulse-swallowing rate. Alternatively, either the N value or the dN input can be set to fractional values to obtain a steady-state, fractional division that would not include the switching dynamics and spurs. In circuit envelope simulation, the divided VCO frequency must remain within its initial envelope bandwidth for all combinations of VCO frequencies and divide numbers. It will not automatically jump from one envelope carrier frequency to another. In transient simulation, because everything is treated as baseband signals, the only constraint is that the time-step must be small enough to cover the maximum frequency.

5. Both the main VCO output and the divided output have an output resistance set by the `Rout` parameter. The main VCO output will deliver the specified Power into a `Rout` load. The divided output will also deliver this amount of power if it is not a baseband output. If it is a baseband output, then the divided output is a sawtooth waveform, whose open circuit voltage represents the instantaneous phase, in radians, of the divided signal. In transient simulation, then, this divided output is always a sawtooth. In circuit envelope, it is a sawtooth if $F0/N$ is within the baseband envelope (it is less than $0.5/\text{timestep}$). If $F0/N$ is closer to one of the analysis carrier frequencies, then the output is a complex sinusoid with the same amplitude as the main VCO output.
6. A Delay parameter value can also be specified for this VCO model. This puts an additional transit delay between the input tuning voltage and the actual change in the output frequency. A delay of at least one timestep does sometimes result in slightly faster simulation speeds, and can be used to model the time delay inherent in any real VCO.
7. Care should be taken when using the VCO in baseband mode. If `F0+VCO_Freq` value goes negative, the model will generate a *negative* frequency, which may give unexpected results. If this is a problem, the `VCO_Freq` expression could include a limiting operator to prevent this.
8. [Figure 5-11](#) shows an example application; simulation results are shown in [Figure 5-12](#). The tuning characteristic is linear in this case and is simply 1 MHz per volt. The nominally divided output frequency is $100\text{MHz}/55=1.81818$ MHz, so this output can be a baseband output, given the 0.1 μsec timestep. The divide number changes from 95 to 55 halfway through the simulation. (Note that while the divided VCO output may not appear to be a regular, uniform, amplitude sawtooth waveform, it does accurately represent the divided signal's phase.)
9. The frequency can be determined by calculating the phase slope; by using linear interpolation, the baseband phase frequency detectors can accurately determine threshold crossings.

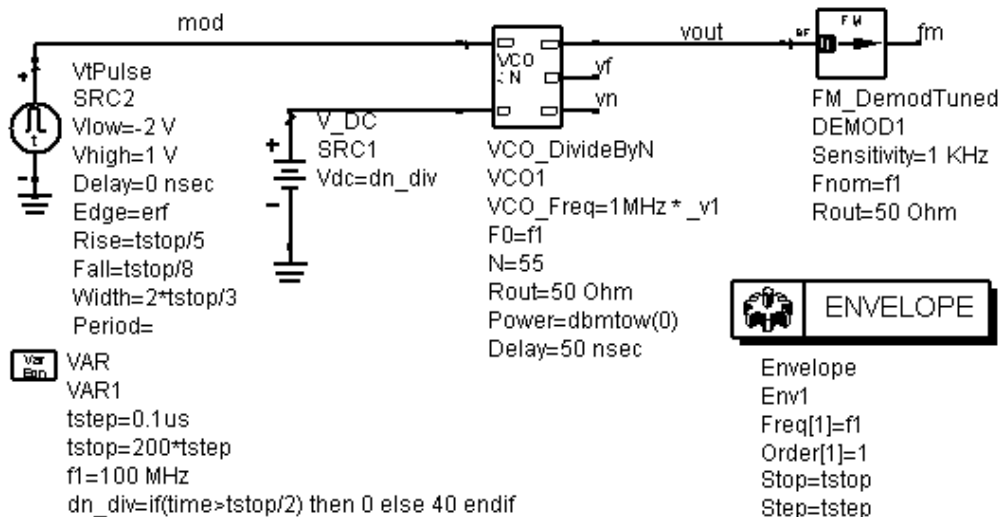


Figure 5-11. Example with Linear Tuning Characteristics

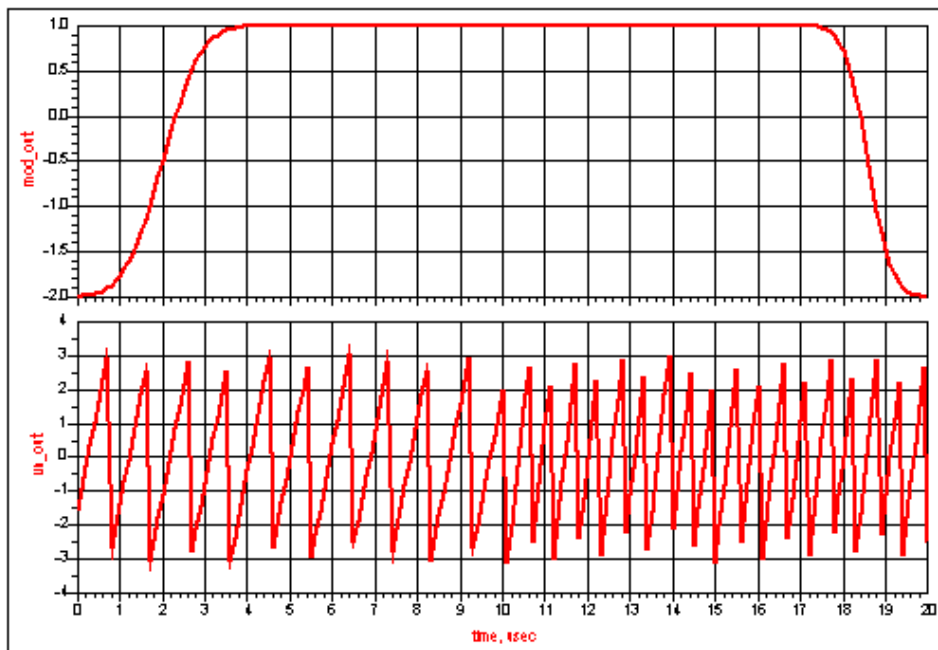
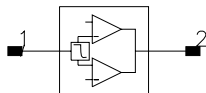


Figure 5-12. Simulation Results

Chapter 6: Switch and Algorithmic Components

Comparator (Comparator)

Symbol



Parameters

Vlow = lower threshold voltage

Vhigh = upper threshold voltage

Notes/Equations

1. Comparator outputs a 1V signal whenever the baseband portion of the input signal is between the two threshold voltages. Input impedance is infinite; output impedance is fixed at 0.1 ohm; and, there is a fixed delay of one timestep.
2. If the baseband portion of V_{in} is greater than Vlow and less than Vhigh, then the output voltage at the next time sample is 1.0V; otherwise, the output is 0.0V.
3. This model works in transient and circuit envelope simulation.

ClockLFSR (Linear Feedback Shift Register)

Symbol



Parameters

Vlow = lower threshold voltage, in volts

Vhigh = upper threshold voltage, in volts

Taps = bits used to generate feedback

Seed = initial value loaded into shift register

Rout = output resistance, in ohms

Notes/Equations

1. ClockLFSR can be used to generate PN sequences with user-defined recurrence relations. The input is a clock signal; with each positive clock edge, the next output bit is calculated. A clock edge occurs any time the baseband input signal rises through 0.5V.
2. This model works in transient and circuit envelope simulation.
3. With each positive clock edge, data is shifted to the right in the shift register. The length of the shift register is determined by the most significant one-bit in the Taps value.

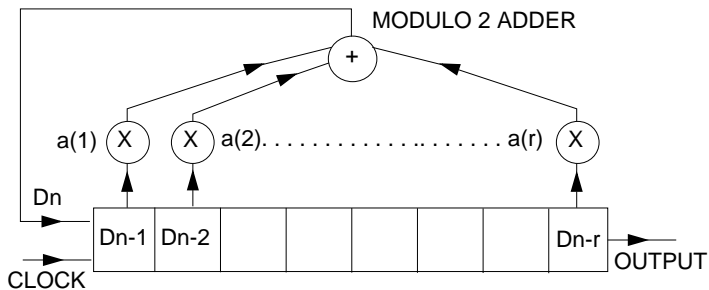


Figure 6-1. LFSR Model

The numbers $a(1), a(2), \dots, a(r)$ are the binary feedback coefficients and are specified by the Taps parameter. This value may be specified as a decimal number, or as a binary pattern if the `bin()` function is used. The initial contents of the shift are specified by the value of the Seed parameter. The following equations describe the operation of this component:

At each positive clock edge n ($n \geq 1$),

for $n \geq 1$:

$$D(n) = \left[\sum_{k=1}^r a(k)D(n-k) \right] \text{mod } 2$$

where

$$D(0) = \text{Seed}_2(0)$$

$$D(-1) = \text{Seed}_2(1)$$

.

.

.

$$D(1-r) = \text{Seed}_2(r-1)$$

$$\text{and Seed} = \sum_{k \geq 0} \text{Seed}_2(k)2^k$$

where $\text{Seed}_2(k) \in \{0,1\}$ for $0 \leq k < r$.

Example: Let $\text{Seed} = 2$, and $\text{Taps} = 7$

then

$$\text{Seed}_2(0) = 0$$

$$\text{Seed}_2(1) = 1$$

.

.

.

$$\text{Seed}_2(6) = 0$$

thus,

$$(0) = \text{Seed}_2(0) = 0$$

$$D(-1) = \text{Seed}_2(1) = 1$$

$$D(-2) = \text{Seed}_2(2) = 0$$

·
·
·

$$D(-6) = \text{Seed}_{22}(6) = 0$$

$$\text{Seed}_2(2) = 0$$

Linear Feedback Shift Register ClockLFSR

The binary feedback coefficients are specified by Taps. For example, the recurrence relation

$$D(n) = (D(n-7) + D(n-3) + D(n-2) + D(n-1)) \bmod 2$$

is specified by:

$$\text{Taps} = \text{bin}("1000111")$$

or

$$\text{Taps} = 71$$

Table 6-1 provides a list of feedback coefficients for linear feedback shift registers showing one or more alternate feedback connections for a given number of stages.

Table 6-1. Feedback Connections for Linear m-Sequences*

No. of Stages	Code Length	Maximal Taps
2	3	[2, 1]
3	7	[3, 1]
4	15	[4, 1]
5	31	[5, 2] [5, 4, 3, 2] [5, 4, 2, 1]
6	63	[6, 1] [6, 5, 2, 1,] [6, 5, 3, 2,]
7	127	[7, 1] [7, 3] [7, 3, 2, 1,] [7, 4, 3, 2,] [7, 6, 4, 2] [7, 6, 3, 1] [7, 6, 5, 2] [7, 6, 5, 4, 2, 1] [7, 5, 4, 3, 2, 1]
8	255	[8, 4, 3, 2] [8, 6, 5, 3] [8, 6, 5, 2] [8, 5, 3, 1] [8, 6, 5, 1] [8, 7, 6, 1] [8, 7, 6, 5, 2, 1] [8, 6, 4, 3, 2, 1]
9	511	[9, 4] [9, 6, 4, 3] [9, 8, 5, 4] [9, 8, 4, 1] [9, 5, 3, 2] [9, 8, 6, 5] [9, 8, 7, 2] [9, 6, 5, 4, 2] [9, 7, 6, 4, 3, 1] [9, 8, 7, 6, 5, 3]
10	1023	[10, 3] [10, 8, 3, 2] [10, 4, 3, 1] [10, 8, 5, 1] [10, 8, 5, 4] [10, 9, 4, 1] [10, 8, 4, 3] [10, 5, 3, 2] [10, 5, 2, 1] [10, 9, 4, 2]
11	2047	[11, 1] [11, 8, 5, 2] [11, 7, 3, 2] [11, 5, 3, 5] [11, 10, 3, 2] [11, 6, 5, 1] [11, 5, 3, 1] [11, 9, 4, 1] [11, 8, 6, 2] [11, 9, 8, 3]

*Reprinted by permission of John Wiley & Sons. From Spread Spectrum Systems, 2nd edition, p. 87, Robert C. Dixon. Copyright © 1984 by John Wiley & Sons, Inc.

Table 6-1. Feedback Connections for Linear m-Sequences* (continued)

No. of Stages	Code Length	Maximal Taps
12	4095	[12, 6, 4, 1] [12, 9, 3, 2] [12, 11, 10, 5, 2, 1] [12, 11, 6, 4, 2, 1] [12, 11, 9, 7, 6, 5] [12, 11, 9, 5, 3, 1] [12, 11, 9, 8, 7, 4] [12, 11, 9, 7, 6, 5] [12, 9, 8, 3, 2, 1] [12, 10, 9, 8, 6, 2]
13	8191	[13, 4, 3, 1] [13, 10, 9, 7, 5, 4] [13, 11, 8, 7, 4, 1] [13, 12, 8, 7, 6, 5] [13, 9, 8, 7, 5, 1] [13, 12, 6, 5, 4, 3] [13, 12, 11, 9, 5, 3] [13, 12, 11, 5, 2, 1] [13, 12, 9, 8, 4, 2] [13, 8, 7, 4, 3, 2]
14	16,383	[14, 12, 2, 1] [14, 13, 4, 2] [14, 13, 11, 9] [14, 10, 6, 1] [14, 11, 6, 1] [14, 12, 11, 1] [14, 6, 4, 2] [14, 11, 9, 6, 5, 2] [14, 13, 6, 5, 3, 1] [14, 13, 12, 8, 4, 1] [14, 8, 7, 6, 4, 2] [14, 10, 6, 5, 4, 1] [14, 13, 12, 7, 6, 3] [14, 13, 11, 10, 8, 3]
15	32,767	[15, 13, 10, 9] [15, 13, 10, 1] [15, 14, 9, 2] [15, 1] [15, 9, 4, 1] [15, 12, 3, 1] [15, 10, 5, 4] [15, 10, 5, 4, 3, 2] [15, 11, 7, 6, 2, 1] [15, 7, 6, 3, 2, 1] [15, 10, 9, 8, 5, 3] [15, 12, 5, 4, 3, 2] [15, 10, 9, 7, 5, 3] [15, 13, 12, 10] [15, 13, 10, 2] [15, 12, 9, 1] [15, 14, 12, 2] [15, 13, 9, 6] [15, 7, 4, 1] [15, 4] [15, 13, 7, 4]
16	65,535	[16, 12, 3, 1] [16, 12, 9, 6] [16, 9, 4, 3] [16, 12, 7, 2] [16, 10, 7, 6] [16, 15, 7, 2] [16, 9, 5, 2] [16, 13, 9, 6] [16, 15, 4, 2] [16, 15, 9, 4]
17	131,071	[17, 3] [17, 3, 2] [17, 7, 4, 3] [17, 16, 3, 1] [17, 12, 6, 3, 2, 1] [17, 8, 7, 6, 4, 3] [17, 11, 8, 6, 4, 2] [17, 9, 8, 6, 4, 1] [17, 16, 14, 10, 3, 2] [17, 12, 11, 8, 5, 2]
18	262, 143	[18, 7] [18, 10, 7, 5] [18, 13, 11, 9, 8, 7, 6, 3] [18, 17, 16, 15, 10, 9, 8, 7] [18, 15, 12, 11, 9, 8, 7, 6]
19	524,287	[19, 5, 2, 1] [19, 13, 8, 5, 4, 3] [19, 12, 10, 9, 7, 3] [19, 17, 15, 14, 13, 12, 6, 1] [19, 17, 15, 14, 13, 9, 8, 4, 2, 1] [19, 16, 13, 11, 19, 9, 4, 1] [19, 9, 8, 7, 6, 3] [19, 16, 15, 13, 12, 9, 5, 4, 2, 1] [19, 18, 15, 14, 11, 10, 8, 5, 3, 2] [19, 18, 17, 16, 12, 7, 6, 5, 3, 1]
20	1, 048,575	[20, 3] [20, 9, 5, 3] [20, 19, 4, 3] [20, 11, 8, 6, 3, 2] [20, 17, 14, 10, 7, 4, 3, 2]
21	2,097,151	[21, 2] [21, 14, 7, 2] [21, 13, 5, 2] [21, 14, 7, 6, 3, 2] [21, 8, 7, 4, 3, 2] [21, 10, 6, 4, 3, 2] [21, 15, 10, 9, 5, 4, 3, 2] [21, 14, 12, 7, 6, 4, 3, 2] [21, 20, 19, 18, 5, 4, 3, 2]
22	4,194,303	[22,1] [22, 9, 5, 1] [22, 20, 18, 16,6, 4, 2, 1] [22, 19, 16, 13, 10, 7, 4, 1] [22, 17, 9, 7, 2, 1] [22, 17, 13, 12, 8, 7, 2, 1] [22, 14, 13, 12, 7, 3, 2, 1]
23	8,388,607	[23, 5] [23, 17, 11, 5] [23, 5, 4, 1] [23, 12, 5, 4] [23, 21, 7, 5] [23, 16, 13, 6, 5, 3] [23, 11, 10, 7, 6, 5] [23, 15, 10, 9, 7, 5, 4, 3] [23, 17, 11, 9, 8, 5, 4, 1] [23, 18, 16, 13, 11, 8, 5, 2]

*Reprinted by permission of John Wiley & Sons. From Spread Spectrum Systems, 2nd edition, p. 87, Robert C. Dixon. Copyright © 1984 by John Wiley & Sons, Inc.

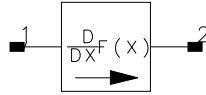
Table 6-1. Feedback Connections for Linear m-Sequences* (continued)

No. of Stages	Code Length	Maximal Taps
24	16,777,215	[24, 7, 2] [24, 4, 3, 1] [24, 22, 20, 18, 16, 14, 11, 9, 8, 7, 5, 4] [24, 21, 19, 18, 17, 16, 15, 14, 13, 10, 9, 5, 4, 1]
25	33,554, 431	[25, 3] [25, 3, 2, 1] [25, 20, 5, 3] [25, 12, 5, 4] [25, 17, 10, 3, 2, 1] [25, 23, 21, 19, 9, 7, 5, 3] [25, 18, 12, 11, 6, 5, 4] [25, 20, 16, 11, 5, 3, 2, 1] [25, 12, 11, 8, 7, 6, 4, 3]
26	67,108,863	[26, 6, 2, 1] [26, 22, 21, 16, 12, 11, 10, 8, 5, 4, 3, 1]
27	134,217,727	[27, 5, 2, 1] [27, 18, 11, 10, 9, 5, 4, 3]
28	268,435,455	[28, 3] [28, 13, 11, 9, 5, 3] [28, 22, 11, 10, 4, 3] [28, 24, 20, 16, 12, 8, 4, 3, 2, 1]
29	536,870,911	[29, 2] [29, 20, 11, 2] [29, 13, 7, 2] [29, 21, 5, 2] [29, 26, 5, 2] [29, 19, 16, 6, 3, 2] [29, 18, 14, 6, 3, 2]
30	1,073,741,823	[30, 23, 2, 1] [30, 6, 4, 1] [30, 24, 20, 16, 14, 13, 11, 7, 2, 1]
31	2,147,483,647	[31, 29, 21, 17] [31, 28, 19, 15] [31, 3] [31, 3, 2, 1] [31, 13, 8, 3] [31, 21, 12, 3, 2, 1] [31, 20, 18, 7, 5, 3] [31, 30, 29, 25] [31, 28, 24, 10] [31, 20, 15, 5, 4, 3] [31, 16, 8, 4, 3, 2]
32	4,294,967,295	[32, 22, 2, 1] [32, 7, 5, 3, 2, 1] [32, 28, 19, 18, 16, 14, 11, 10, 9, 6, 5, 1]
33	8,589,934,591	[33, 13] [33, 22, 13, 11] [33, 26, 14, 10] [33, 6, 4, 1] [33, 22, 16, 13, 11, 8]
61	2,305,843,009,213, 693, 951	[61, 5, 2, 1]
89	618,970,019,642,690, 137,449,562,112	[89, 6, 5, 3]

*Reprinted by permission of John Wiley & Sons. From Spread Spectrum Systems, 2nd edition, p. 87, Robert C. Dixon. Copyright © 1984 by John Wiley & Sons, Inc.

Differentiator (Differentiator)

Symbol



Parameters

Gain = differentiator slope

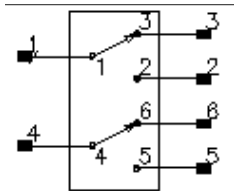
Rref = reference resistance for both ports, in ohms

Notes/Equations

1. The output voltage is equal to the derivative (with respect to time) of the input voltage.
2. The Gain parameter is entered as a linear quantity, not in dB.
3. Differentiator works in both transient and envelope simulations.
4. The input resistance of the integrator is Rref. The output is a voltage source Vout in series with output resistance Rref.

DPDT_Static (Double Pole Double Throw Switch, Static)

Symbol



Parameters

State = state of switch: 0 (nodes 1 and 2, 4 and 5 connected); 1 (nodes 1 and 3, 4 and 6 connected)

F1 = first frequency breakpoint

F2 = second frequency breakpoint

F3 = third frequency breakpoint

Loss1 = attenuation for frequencies $\leq F1$, in dB

Loss2 = attenuation for frequencies $>F1 \leq F2$, in dB

Loss3 = attenuation for frequencies $>F2 \leq F3$, in dB

Loss4 = attenuation for frequencies $>F3$, in dB

VSWR1 = VSWR at both ports for frequencies $\leq F1$

VSWR2 = VSWR at both ports for frequencies $>F1 \leq F2$

VSWR3 = VSWR at both ports for frequencies $>F2 \leq F3$

VSWR4 = VSWR at both ports for frequencies $>F3$

Isolat = isolation, in dB

ZRef = reference impedance for all ports, in ohms

Temp = temperature, in degree C

Notes/Equations

1. This model works in transient and circuit envelope simulation.
2. Up to three frequency break points can be used to define four bands, with different losses and VSWR for each frequency band. Losses are entered as a positive attenuation. In other words, a 0.5 dB loss is entered as $L1=0.5$, not $L1=-0.5$.

Isolation is assumed to be constant across all frequency bands.

When using this device in a transient simulation, the switch setting must be constant.

If loss values are specified versus frequency via a dataset and using the component DATASETVARIABLE, then a time delay through the device will be introduced when using transient analysis.

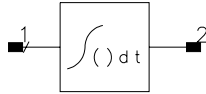
Transient Analysis Time Delay = $(1/(2 \times \text{MAX_FREQUENCY})) \times \text{POINTS}$

where

$$\begin{aligned} \text{POINTS} &= (\text{FILTER_ORDER}/2) - 1 \quad \text{for FILTER_ORDER even} \\ &= (\text{FILTER_ORDER} + 1)/2 - 1 \quad \text{for FILTER_ORDER odd} \end{aligned}$$

IntegratorSML (Integrator)

Symbol



Parameters

GainAC = integrator gain

GainDC = gain of dc constant

Rref = reference resistance for both ports, in ohms

Notes/Equations

1. The output voltage is equal to the integral (with respect to time) of the input voltage. This device has been modeled after a physically realizable integrator. The dc gain is not infinite and is specified by GainDC.
2. GainAC is the ac gain of the integrator (typically set to 1.0); GainDC is the dc gain of the integrator. Both of these are a linear quantity, not in dB.
3. An ideal integrator has infinite gain at dc, which is not physically realizable. These parameters describe the operation of the integrator that has been modeled after a traditional high gain differential amplifier with capacitive feedback and a series input resistor. GainDC is the open loop dc gain of the amplifier in this model. Gain is equal to $1/RC$ where R is the input series resistor value in ohms and C is the feedback capacitor value in farads.
4. The integrator transfer function follows:

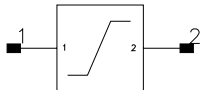
$$\frac{V_{out}}{V_{in}} = \frac{GainDC}{1 + (j \times w \times GainAC \times GainDC)}$$

where w is the frequency in radians/sec.

5. IntegratorSML works in both transient and envelope simulations.
6. The input resistance of the integrator is Rref. The output is a voltage source Vout in series with output resistance Rref.

LimiterSML (Limiter)

Symbol



Parameters

Gain = linear gain

Vmax = maximum output voltage, in volts

Vmin = minimum output voltage, in volts

Rout = output resistance, in ohms

Fnom = nominal input frequency, in hertz

Range of Usage

$F_{nom} \geq 0$

Notes/Equations

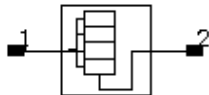
1. The Gain parameter is a real value only (not in dB or complex).
2. This model can be used in transient, harmonic balance and circuit envelope simulations.
3. When used in transient simulations, Fnom must be set to zero. Gain, Vmin and Vmax determines the output voltage time series. The output voltage time series is calculated as Gain times the input voltage time series but is hard limited to Vmin and Vmax in case the value drops below Vmin or rises above Vmax. Hard limiting is not allowed to drop the magnitude of the output voltage by more than a factor of 10^{-4} at any point in time. If this happens, the input signal is simply scaled by Gain and 10^{-4} . Thus, a sine wave with 10^6 V amplitude as input to a LimiterSML component with Gain=10, Vmax=1V and Vmin=-1V will scale to an output signal with a 10^3 V amplitude, not a 1V amplitude. This signal will be added to a pulse function that is Vmin for negative input signals and Vmax for positive input signals so the output will not be a perfect sine wave.
4. When used in harmonic balance simulations, Fnom must be set to a non-zero fundamental or intermodulation frequency. Fnom, Gain and $\text{abs}(V_{\text{max}})$ determines the output voltage spectrum (Vmin is not used). The output voltage spectrum is calculated as Gain times the input voltage spectrum but is scaled

by $\min(\text{abs}(V_{\text{max}})/\text{abs}(V(F_{\text{nom}})), 1)$ at all frequencies, where $V(F_{\text{nom}}) = \text{Gain} \times V_{\text{in}}(F_{\text{nom}})$. This gain limits the entire frequency spectrum by $\text{abs}(V_{\text{max}})/\text{abs}(V(F_{\text{nom}}))$ in the case where $\text{abs}(V(F_{\text{nom}}))$ exceeds $\text{abs}(V_{\text{max}})$. If there is no spectral component at F_{nom} , no gain limiting is performed. No minimum gain of 10^{-4} applies in this case.

5. When used in circuit envelope simulations, F_{nom} can be a zero or non-zero fundamental or intermodulation frequency.
 - If F_{nom} is zero, the mode of operation is similar to that for transient analysis. If there is a baseband signal, it is hard limited as dictated by V_{min} and V_{max} . Hard limiting is not allowed to drop the magnitude of the output voltage by more than a factor of 10^{-4} at any time (refer to note 3). All other spectral components are scaled accordingly. If there is not a baseband signal, nothing happens.
 - If F_{nom} is a non-zero fundamental or intermodulation frequency, the mode of operation is similar to that for harmonic balance analysis. All spectral components, including the baseband signal if applicable, are gain limited as dictated by $\min(\text{abs}(V_{\text{max}})/\text{abs}(V(F_{\text{nom}})), 1)$. For harmonic balance analysis, the value of V_{min} does not matter in this case.
6. LimiterSML has more than one mode of operation. It performs clipping or limiting in a Transient or Circuit Envelope analysis, and it performs gain scaling on the spectral components in a Harmonic Balance analysis. Its mode of operation depends on the value of F_{nom} and the type of simulation being performed.

ParallelSerial (Parallel to Serial Shift Register)

Symbol



Parameters

OutputRate = serial output data clock rate, in hertz

LSB_First = output serial data with least significant bit first = YES; NO = output serial data with most significant bit first

Delay = initial synchronization delay

InputBits = number of bits in input word

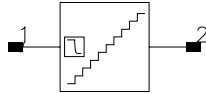
IntegerIn = scale input data as integers: YES, NO

Notes/Equations

1. This Parallel To Serial Shift Register model is used to convert a sequence of input words into a serial output bit stream. The fixed serial output bit rate is specified by the OutputRate parameter. The serial data can be output with either the LSB or MSB first, depending on the state of the LSB_First parameter. The output impedance is fixed at 0.1 ohm. A logic one generates an open circuit voltage of 1.0V. A logic zero is 0.0V.
2. The number of bits in each input word is specified by InputBits. The input is sampled at a rate equal to OutputRate/InputBits, with an initial synchronization delay specified by Delay.
3. The input impedance is infinite. If the IntegerIn parameter is set true, then the input is assumed to be scaled as an integer from 0.0V to $2^{\text{OutputRate_1V}}$. Otherwise, the input is assumed to be scaled from -1.0 to +1.0V, with -1.0V interpreted at word 0.
4. This model works in transient and circuit envelope simulation. Only the baseband portion of circuit envelope voltage is used.

QuantizerSML (Quantizer)

Symbol



Parameters

V_{min} = minimum baseband input voltage, in voltage units

V_{max} = maximum baseband input voltage, in voltage units

N = number of quantized output levels

OutState = output is an integer representing quantization level = nonzero; output is scaled same as input but is quantized = zero

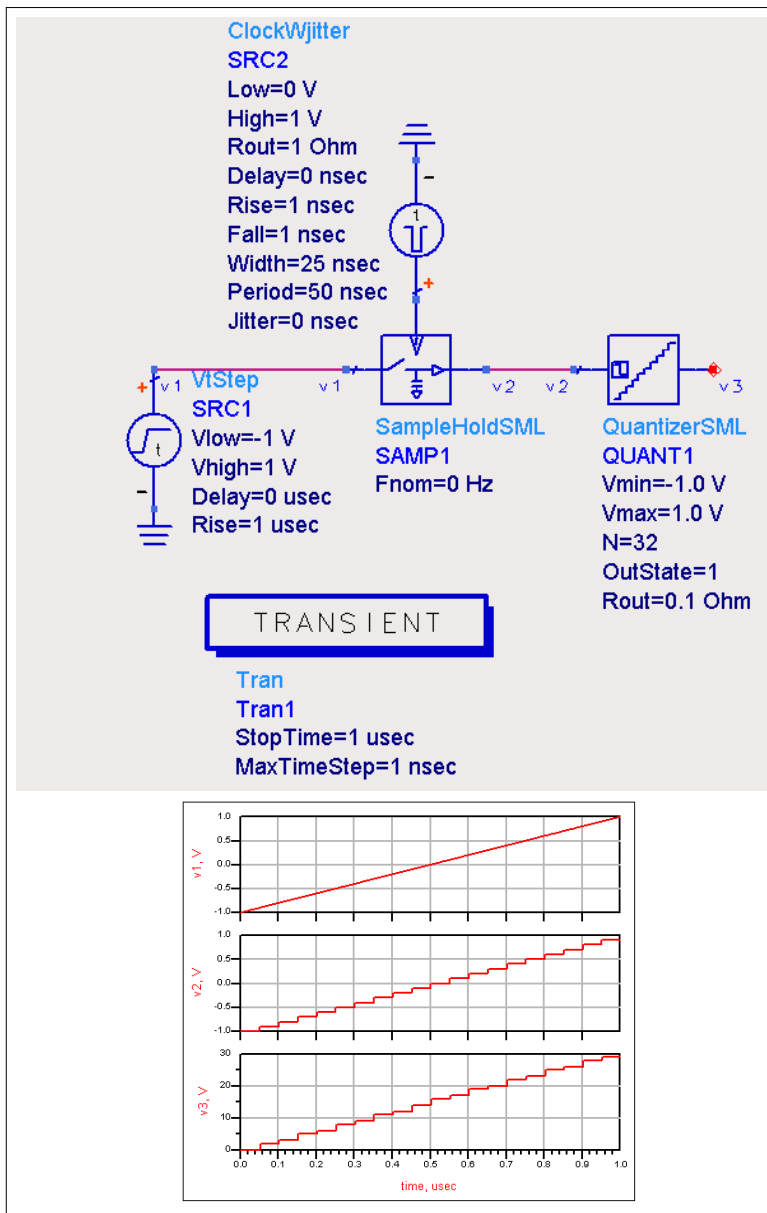
Rout = output resistance, in ohms

Notes/Equations

1. This quantizer model outputs a delayed, uniformly quantized representation of the baseband portion of the input voltage. The input impedance is infinite. The delay is fixed at one timestep. The quantization state output mode is in offset binary format.

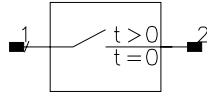
The input quantization level is equal to $\Delta = (V_{max} - V_{min}) / (N - 1)$. Any input less than $(V_{min} + \Delta/2)$ is assigned to state 0. Any input greater than $(V_{max} - \Delta/2)$ has a state equal to $(N - 1)$. If the OutState parameter is true, this quantization state is directly output. If the OutState parameter is equal to zero, the quantization state is scaled and offset, $V_{out} = Q/\Delta + V_{min}$, to match the input voltage.

2. This model works in transient and circuit envelope simulation.
3. An example of using this component along with SampleHoldSML follows.



ResetSwitch (Reset Switch)

Symbol



Parameters

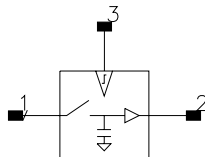
None

Notes/Equations

1. This reset switch is an ideal switch that is closed at time=0 and open for time>0. Its main use is in time domain simulation modes (transient and circuit envelope) to allow the resetting of components such as ideal integrators. Often these blocks that have infinite gain at dc do not allow the circuit simulator to converge, but by keeping them reset at time=0, a valid initial solution can be obtained prior to the actual time domain simulation and then the switch can be opened to observe the time domain response. In steady-state analysis, such as dc or harmonic balance, time is always 0 so this switch remains closed.

SampleHoldSML (Sample Hold)

Symbol



Parameters

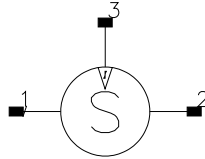
F_{nom} = nominal input and output frequency, in hertz

Notes/Equations

1. This model samples the input signal and holds it until the next sample event. It is a tuned model; it selects and holds just the signal at the input harmonic closest to the frequency specified by F_{nom} . If there is no analysis harmonic frequency close enough to the F_{nom} frequency, a warning is issued and the output is 0. The input impedance of both the sample clock input and the signal input is infinite. The output is a voltage source with a fixed impedance of 0.1 ohm.
2. The sampling instant is defined to be when the sample clock input rises through the 0.5V fixed threshold. If the baseband mode of operation is specified ($F_{nom} \sim 0$ Hz), this behaves as a normal sample and hold; the baseband input voltage at the sampling instant is output and then held indefinitely at this constant value until the next sample. If an envelope carrier frequency is specified by F_{nom} , the signal component at this envelope harmonic frequency is selected and its complex value at the sampling instant is held at the output. This held output appears at the same envelope harmonic frequency. For example, if an envelope analysis with a fundamental of 1 GHz is performed with a timestep of 0.1nsec and an offset frequency source at 1.001 GHz is sampled, the complex value of the envelope is sampled and held constant. The output will appear to be at the 1 GHz fundamental frequency since the complex envelope is no longer changing, until the next sample event.
3. The Sampler model samples the entire spectrum and outputs just a real baseband voltage.
4. SampleHoldSML works in transient and circuit envelope simulation. In transient simulation, F_{nom} has no effect because all signals are baseband only.

Sampler (Sampler)

Symbol



Parameters

Ton = ON-state pulse width, switch in low impedance state, in seconds

Ron = ON-state resistance, switch in low impedance state, in ohms

S11 = input reflection coefficient

Z0 = input reference impedance, in ohms

Notes/Equations

1. Sampler is a linear behavioral model of a high-frequency sampler. It can also be used to model the sampling efficiency and droop of lower frequency sample-holds.
2. This model works in transient and circuit envelope simulation. In envelope simulation, the input signal is determined by transforming all the spectral input voltages at the sampling instant to determine the total, real instantaneous voltage. Prior to being sampled, the input signal is first filtered with an ideal sinc() filter determined by the Ton pulse width. The input match can be complex, but since it needs to represent a causal response it cannot be complex at dc.
3. The input impedance looking into the sampler is determined by Z0 and S11 with the standard formula:

$$Z_{in} = \frac{Z0(S11 + 1)}{S11 - 1}$$

Usually, S11=0 and Z0=50 results in the input impedance of the sampler being 50 ohms.

4. The Clock/LO input impedance is infinite, and the sampling instant is defined only by the baseband portion of the input signal and occurs whenever the baseband signal passes through 0.5V with a positive slope.

5. Sampler has two basic modes of operation. If R_{on} is equal to 0, then the output is a ideal voltage source with a value equal to the last sampled value of the filtered input voltage; if R_{on} is not equal to 0, then the output impedance is time-varying. In the quiescent state, it is $1T_{ohm}$. However when a sample occurs, the impedance for that time point is reduced to a sampler resistance equal to $R_{on} \times \text{timestep} / T_{on}$. The actual sampler efficiency is then determined by sampler R_{on} and T_{on} values, and the load capacitance on the sampler output.

If $T_{on} \gg R_{on} \times C_{Load}$, the sampler will behave with 100% efficiency. In a microwave sampler, this efficiency is typically much less than 100%. The hold time constant in this sampler mode is determined strictly by the capacitive and resistive load placed on the sampler output by the user.

The R_{on} sampler parameter should include all charging impedances, including the sampler switch impedance as well as the effect of any source impedance.

The output signal in Envelope mode is a baseband-only signal. No RF leakage is included in this model.

Note that due to the impulse nature of this sampler model output, the analysis integration order should be set to 1 (Backward Euler) when using this model. This is especially true when sampling rapidly changing input signals and when the sampler parameters are set for high sampling efficiency.

6. In the circuit envelope example in [Figure 6-2](#), the input signal is a high-frequency sawtooth waveform with 15 harmonics. Its frequency is set 10 kHz above 1 GHz. The sampler is being driven at a 500 kHz rate so the output signal should be a replicated sawtooth at 10 kHz. Simulation results for two different combinations of R_{on} and T_{on} are shown in [Figure 6-3](#). The high-efficiency mode tracks the input. The low-efficiency mode (50 ohm, 50 psec) shows its lowpass filtering impact. Note the two different time scales: 2 nsec for the input trace and 200 μ sec for the output traces.

The example in [Figure 6-4](#) shows a low-frequency application that has a significant droop due to the finite output resistance. Simulation results are shown in [Figure 6-5](#).

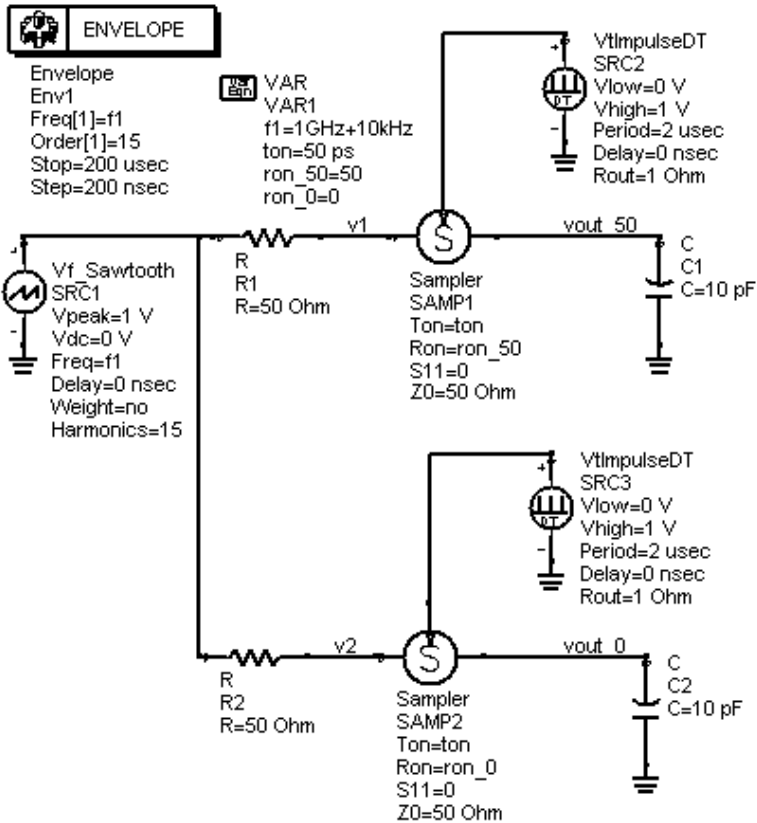


Figure 6-2. Circuit Envelope Example,
High-Frequency Application

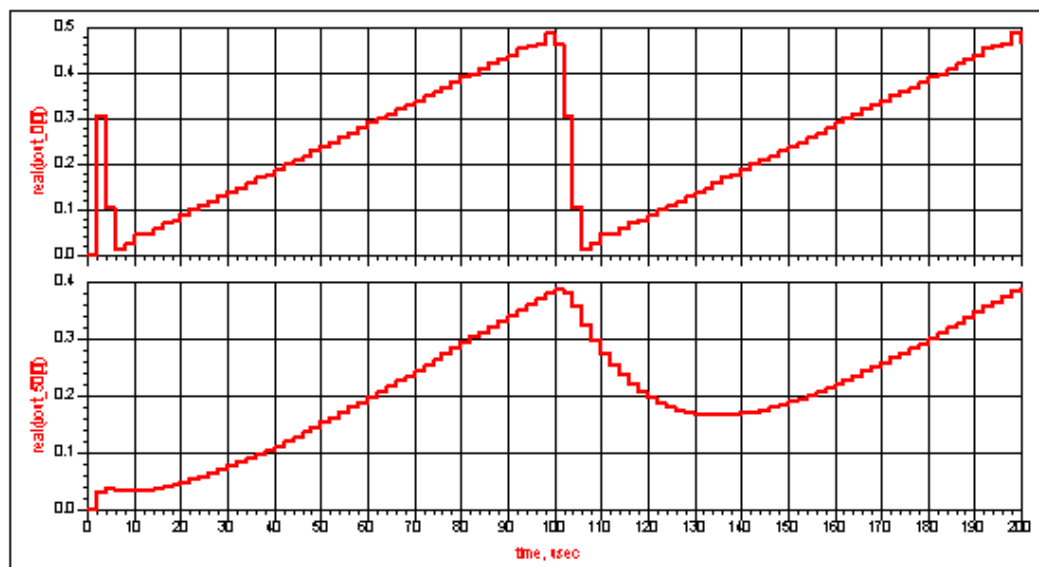


Figure 6-3. Simulation Results

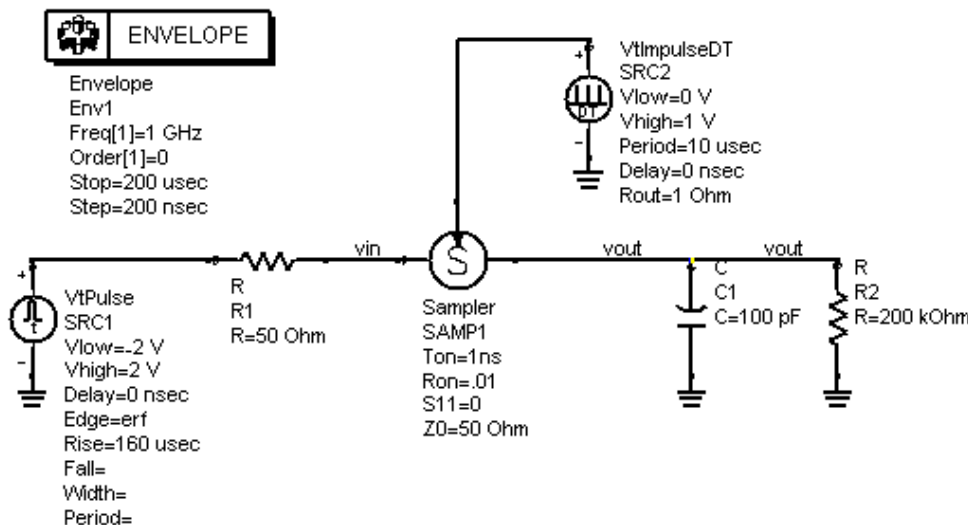


Figure 6-4. Circuit Envelope Example, Low-Frequency Application

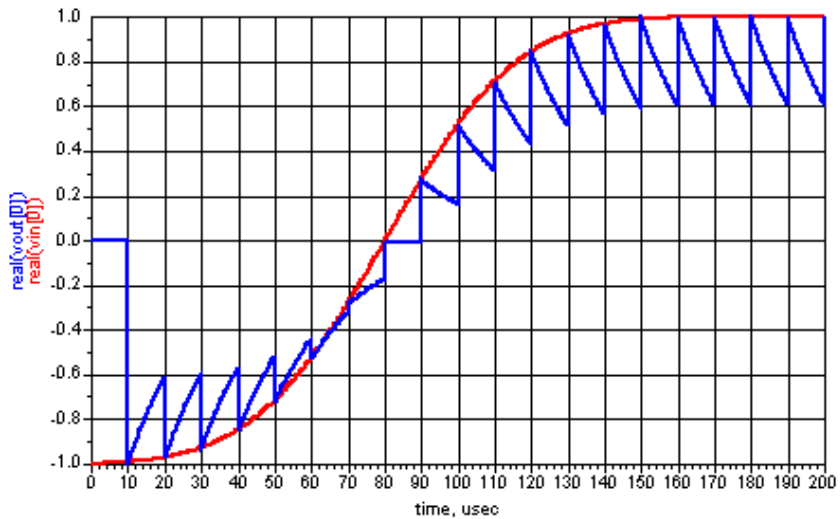


Figure 6-5. Simulation Results

SerialParallel (Serial to Parallel Shift Register)

Symbol



Parameters

InputRate = serial input data clock rate, in hertz

LSB_First = serial data arrives with least significant bit first = YES; NO = most significant bit arrives first

Delay = initial synchronization delay, in seconds

OutputBits = number of bits in output word

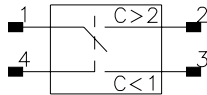
IntegerOut = scale output data as integers (instead of from -1 to 1): YES, NO

Notes/Equations

1. This SerialParallel shift register model is used to convert a serial data stream into a multiple bit word. The fixed serial input bit rate is specified by the InputRate parameter, with an initial synchronization delay specified by the Delay parameter. The serial data can be interpreted as either LSB or MSB first, depending on the state of the LSB_First parameter. The input impedance is infinite. A voltage less than $0.5V$ is a logical zero.
2. The number of bits in each output word is specified by OutputBits. The output is updated at a rate equal to $\text{InputRate}/\text{OutputBits}$. The open circuit output voltage is an integer value from 0 to $2^{\text{OutputBits}-1}$, if the IntegerOut parameter is true. Otherwise, the output is scaled from -1 to $+1V$, with all 0 s corresponding to $-1V$ and all 1 s to $+1V$ s. The output impedance is fixed at 0.1 ohm .
3. This model works in transient and circuit envelope simulation. Only the baseband portion of circuit envelope voltage is used.

SPDT_Dynamic (Single Pole Double Throw Switch, Dynamic)

Symbol

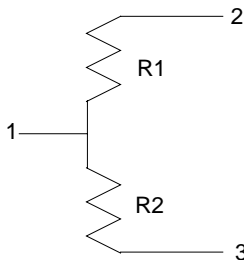


Parameters

R_{on} = on-state resistance of switch, in ohms

R_{off} = off-state resistance of switch, in ohms

Range of Usage



Pin 1 is the input, pins 2 and 3 are output, pin 4 is the control voltage, V_c .

If $V_c > 2V$, $R_1 = R_{on}$, $R_2 = R_{off}$

If $V_c < 1V$, $R_1 = R_{off}$, $R_2 = R_{on}$

When V_c increases from 1V to 1.5V, R_2 changes from R_{on} to R_{off}

When V_c increases from 1.5V to 2V, $R_2 = R_{off}$ and R_1 changes from R_{off} to R_{on}

At $V_c = 1.5V$, $R_1 = R_2 = R_{off}$

Notes/Equations

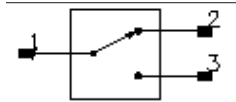
1. This SPDT_Dynamic switch model, as opposed to the SPDT_Static model, can be used to dynamically switch states in response to the input control voltage. The input impedance of the control voltage port is infinite. Whenever the control voltage is greater than 2.0V, the input is connected to the (c>2) output with a resistance equal to R_{on} ; otherwise, there is a resistance of R_{off} ohms between the two pins. Similarly, whenever the control voltage is less than 1.0V, the input is connected to the (c<1) output with a resistance of R_{on} ; otherwise, there is a resistance of R_{off} between the two pins. Note that when the control

voltage is between 1.0V and 2.0V, the switch is open with respect to both outputs, simulating the break operation before making connection.

2. Capacitances, leakage currents, or other non-idealities are not included in this model.
3. This model is primarily meant for usage in the time domain (transient and circuit envelope) simulation with baseband control voltages. Due to the instantaneous, abrupt nature of the switching action, driving the control input with a carrier frequency input, in either harmonic balance or circuit envelope, should be avoided or done with careful consideration of the number of harmonics required and potential convergence problems.

SPDT_Static (Single Pole Double Throw Switch, Static)

Symbol



Parameters

State = state of switch: for 0, nodes 1 and 2 are connected; for 1, nodes 1 and 3 are connected

F1 = first frequency breakpoint

F2 = second frequency breakpoint

F3 = third frequency breakpoint

Loss1 = attenuation for frequencies $\leq F1$, in dB

Loss2 = attenuation for frequencies $>F1 \leq F2$, in dB

Loss3 = attenuation for frequencies $>F2 \leq F3$, in dB

VSWR1 = VSWR at both ports for frequencies $\leq F1$

VSWR2 = VSWR at both ports for frequencies $>F1 \leq F2$

VSWR3 = VSWR at both ports for frequencies $>F2 \leq F3$

Isolat = isolation, in dB

ZRef = reference impedance for all ports, in ohms

Temp = temperature, in degree C

CheckPassivity = check passivity flag: if set to yes (default), a passivity check is performed and a warning is output if the device is not passive; if set to no, a passivity check is not performed.

Notes/Equations

1. This model is based on S-parameters ($S12 = S21$, $S13 = S31$).
2. Up to three frequency break points can be used to define four bands, with different losses and VSWR for each frequency band.

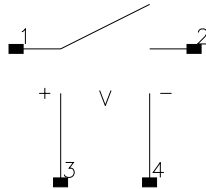
Enter loss as a positive attenuation. In other words, a 0.5 dB loss is entered as Loss = 0.5, not Loss = -0.5.

3. Isolation is assumed to be constant across all frequency bands.

When using this device in a transient simulation, the switch setting must be constant.

SwitchV (Voltage Controlled Switch)

Symbol



Parameters

Model = name of a SwitchV_Model

R1 = resistance at voltage 1, in ohms

V1 = voltage 1, in volts

R2 = resistance at voltage 2, in ohms

V2 = voltage 2, in volts

Range of Usage

$R1, R2 > 0$

$V1 \neq V2$

Notes/Equations

1. This component implements a voltage controlled switch. The switch resistance varies as a function of the applied control voltage:

$$R_{SW} = \begin{cases} R1 & \text{if } V_{CON} \leq V1 \\ f(V_{CON}) & \text{if } V1 \leq V_{CON} \leq V2 \\ R2 & \text{if } V_{CON} \geq V2 \end{cases}$$

where

$$f(V_{CON}) = \begin{cases} x = \frac{V_{CON} - V1}{V2 - V1} \\ \text{if } x < 1/2 \\ y = 2x^2 \\ \text{else} \\ y = 1 - 2(x-1)^2 \\ \ln(R_{SW}) = \ln(R1) + y \times \ln(R2/R1) \end{cases}$$

For these equations, it is assumed that $V1 < V2$. If $V2 < V1$, swap $R1$ and $V1$ with $R2$ and $V2$ in the equation above.

V_{CON} is defined so that $f(V1) = R1$ and $f(V2) = R2$ and provides a smooth transition between states. This is a plot of the switch resistance as a function of the control voltage for $V1=1$, $R1=10$, $V2=3$, $R2=10000$.

2. The Model name is optional. If the model name is not specified, then $R1$, $V1$, $R2$ and $V2$ must all be specified on the SwitchV instance; if the model name is specified, then $R1$, $V1$, $R2$ and $V2$ on the SwitchV instance are optional and override the values specified in the SwitchV_Model.
3. $R1$ and $R2$ are model parameters, not the resistance at nodes. Convergence problems may occur if the difference between $R1$ and $R2$ is too large.
4. $V1$ and $V2$ are model parameters, not node voltages. Convergence problems may occur if the difference between $V1$ and $V2$ is too small.
5. This component works in all analyses, including transient and circuit envelope simulation.

If a SwitchV_Model item is not present, the SwitchV Model parameter should be blank.

6. [Table 6-2](#) lists the DC operating point parameters that can be sent to the dataset.

Table 6-2. DC Operating Point Information

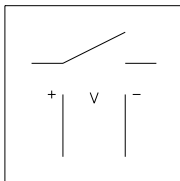
Name	Description	Units
Is	Current	A

Table 6-2. DC Operating Point Information

Name	Description	Units
Power	DC power dissipated	W
Vs	Voltage	V

SwitchV_Model (Voltage Controlled Switch Model)

Symbol



Parameters

R1 = resistance at voltage 1, in ohms

V1 = voltage 1, in volts

R2 = resistance at voltage 2, in ohms

V2 = voltage 2, in volts

AllParams = DataAccessComponent-based parameters

Range of Usage

$R1, R2 > 0$

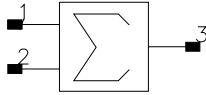
$V1 \neq V2$

Notes/Equations

1. This model supplies values for a SwitchV component.
2. The range between R1 and R2 should not be too large, or convergence problems may occur.
3. The difference between V1 and V2 should not be too small, or convergence problems may occur.
4. R1, V1, R2 and V2 can be optionally specified on the SwitchV instance that uses this model; those values override the model values.
5. This model works in transient and circuit envelope simulation.
6. Use AllParams with a DataAccessComponent to specify file-based parameters (refer to the *DataAccessComponent*). Note that model parameters that are explicitly specified take precedence over those specified via AllParams.

VSum (Voltage Summer)

Symbol



Parameters

None

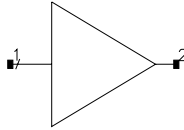
Notes/Equations

1. This component has infinite input impedance and zero output impedance. Output voltage is equal to voltage summed from the two input ports.

Chapter 7: Tx/Rx Subsystems

RF_PA_CKT (RF Power Amplifier Circuit)

Symbol



Parameters

None

Notes/Equations

1. This is a 2-stage BJT amplifier with the center frequency of 2GHz, small signal gain of ~30dB, and maximum power output of ~15dBm. This circuit is provided for the convenience of easy inclusion of a circuit level amplifier in an Agilent Ptolemy Envelope cosimulation.

RF_RX_SML (RF Receiver)

Symbol



Parameters

RX_AntTemp = Receiving antenna noise temperature, in Kelvin (default: 150)

RX_Gain = Receiver gain, in dB (default: 80 dB)

RX_NF = Receiver noise figure in dB (default: 5 dB)

RF_Freq = RF frequency, in MHz, GHz, or THz (default: 900 MHz)

RF_BW = RF bandwidth, in Hz, kHz, MHz, or GHz (default: 25 MHz)

IF_Freq1 = First IF frequency in Hz, kHz, MHz, or GHz (default: 100 MHz)

IF_Freq2 = Second IF frequency in Hz, kHz, MHz, or GHz (default: 400 KHz)

IF_BW = IF bandwidth in Hz, kHz, MHz, or GHz (default: 30 KHz)

IP3in = Receiver input IP3, in pW, nW, uW, mW, W, kW, dBm (default: -25 dBm)

RIn = Input resistance in Ohm, mOhm, kOhm, MOhm, GOhm (default: 50 Ohm)

ROut = Output resistance in Ohm, mOhm, kOhm, MOhm, GOhm (default: 50 Ohm)

Notes/Equations

1. This is a super-heterodyne RF receiver subsystem. Input is an RF signal; output is an IF signal. Three CW spurs are combined with the input RF signal to simulate the receiver interference performance.

This subnetwork is provided for easy inclusion of an RF receiver in an Agilent Ptolemy Envelope co-simulation.

RF_TX_SML (RF Transmitter)

Symbol



Parameters

IF_Freq = IF frequency in Hz, kHz, MHz, or GHz (default: 400 kHz)

RF_Freq = RF frequency, in Hz, kHz, MHz, or GHz (default: 900 MHz)

RF_BW = RF bandwidth, in Hz, kHz, MHz, or GHz (default: 30 KHz)

TX_Gain = Transmitter gain, in dB (default: 80 dB)

PSat = Saturated power at output, in dBm, dBW, nW, uW, mW, W, or kW (default: 35 dBm)

RIn = Input resistance in Ohm, mOhm, kOhm, MOhm, GOhm (default: 50 Ohm)

ROut = Output resistance in Ohm, mOhm, kOhm, MOhm, GOhm (default: 50 Ohm)

Notes/Equations

1. This is an RF transmitter subsystem. Input is an IF signal; output is an RF signal.

This subnetwork is provided for easy inclusion of an RF transmitter in an Agilent Ptolemy Envelope co-simulation.

Chapter 8: System Data Models

Introduction

The *Filters* - *<filter type>* and *System* - *<device type>* palettes contain two fundamentally different types of behavioral system models.

Filters, *System - Amps & Mixers*, and *System - Mod/Demod* can be classified as *tops-down system models* that support a tops-down system design flow where model behaviors are characterized by a small number of independent parameters such as frequency, power and load. They are often referred to as *parameter-based behavioral models*.

System - Data Models can be classified as *bottoms-up system models* that support a bottoms-up verification flow where model behaviors are extracted from a simulation (or measurement) of a transistor-level circuit. They are often referred to as *data-based behavioral models*.

The *parameter-based behavioral models* typically provide superior speed relative to the data-based behavioral models with both of these being vastly superior to a brute-force transistor-level simulation.

The *data-based behavioral models* typically provide superior accuracy relative to the *parameter-based behavioral models* as they capture actual behaviors of implemented circuit components and not just design specifications.

The differences between *parameter-* and *data-based behavioral models* justify a palette emphasis on flow (all data-based behavioral models grouped together) rather than functionality (all amplifiers, mixers, modulators, and demodulators grouped together).

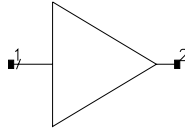
The use model for parameter and data-based behavioral models is slightly different. For parameter-based models, a series of parameters must be set prior to using the model. For data-based models, circuit characteristics must be extracted prior to using the model. Typically, the *<DeviceName>_Setup* component is an extractor that does the initial circuit characterization and generates dataset or file while the *<DeviceName>_Data* component is a corresponding model that uses the dataset or file generated by *<DeviceName>_Setup*. These components are placed next to each other on the palette. Note that a few models have a slightly different use model and that the naming convention is not consistent across all models.

To support *System-Data Models* components, the *BehavioralModels* example suite was developed that contains a series of examples demonstrating the use of the different data-based system models; it provides a good starting point for achieving familiarity with the various data-based system models.

In addition to the data-based behavioral model components, the *System-Data Models* palette contains Balun3Port and Balun4Port; these components are also accessible from the *System-Passive* palette.

AmpH1H2 (Amplifier/Fundamental and 2nd Harmonic vs. Input Power)

Symbol



Parameters

Dataset = Name of dataset containing data for this amplifier model (can be file-based)

G1expr = Gain expression for fundamental frequency

G2expr = Gain expression for second harmonic frequency

SP11 = Forward reflection coefficient; use $x+j*y$, polar (x, y), dbpolar (x, y), vswrpolar (x,y) for complex value

SP22 = Reverse reflection coefficient; use $x+j*y$, polar (x, y), dbpolar (x, y), vswrpolar (x,y) for complex value

SP12 = Reverse transmission coefficient; use $x+j*y$, polar (x, y), dbpolar (x, y), vswrpolar (x,y) for complex value

NF = Noise figure in dB

NFmin = Minimum noise figure at S_{opt} in dB

S_{opt} = Optimum source reflection for minimum noise figure; use $x+j*y$, polar (x, y), dbpolar (x, y), vswrpolar (x,y) for complex value

Rn = Equivalent noise resistance, in mOhm, Ohm, kOhm, MOhm, or GOhm

Z1 = Reference impedance for Port1 in mOhm, Ohm, kOhm, MOhm, or GOhm

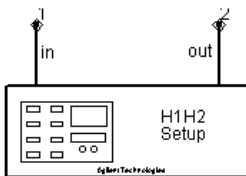
Z2 = Reference impedance for Port2 in mOhm, Ohm, kOhm, MOhm, or GOhm

Notes/Equations

1. AmpH1H2 is a data-based system model of a circuit-level amplifier. The circuit-level amplifier is characterized by a dataset generated by the system component AmpH1H2_Setup.
2. Through G1expr and G2expr, AmpH1H2 models both odd and even harmonics.
3. An example of the use of this device is provided; access the example from the ADS Main window: *File > Example Project > BehavioralModels > AmpH1H2_prj*.

AmpH1H2_Setup (Amplifier/Fundamental and 2nd Harmonic vs. Input Power Setup)

Symbol



Parameters

Freq = Frequency, in Hz

Order = Order for Harmonic Balance analysis, integer

Pin_Start = Input power start value, in dBm

Pin_Stop = Input power stop value, in dBm

Pin_Step = Input power step value, in dB

Range of Usage

Freq ≥ 0

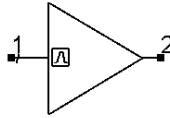
Order > 0

Notes/Equations

1. AmpH1H2_Setup performs a swept input power Harmonic Balance simulation of a circuit-level amplifier and generates a dataset for subsequent use by the data-based system model AmpH1H2.
2. AmpH1H2_Setup can be pushed into for a view of the implementation. If necessary, the component can be copied and modified to suit individual needs.
3. In order for AmpH1H2 to produce an accurate model of the circuit-level amplifier characterized via AmpH1H2_Setup, it is necessary that the Order parameter for AmpH1H2_Setup is large enough to prevent aliasing of higher-order frequency components. For mildly non-linear circuits, an Order of around five should suffice while an Order in the range 10-15 is recommended for highly non-linear circuits.
4. An example of the use of this device is provided; access the example from the ADS Main window: *File > Example Project > BehavioralModels > AmpH1H2_prj.*

AmplifierP2D (P2D File Amplifier; FDD-Based, for Single Carrier Signal)

Symbol



Parameters

Freq = Nominal input frequency, in Hz

P2DFile = Filename for P2D data

Notes/Equations

1. AmplifierP2D is a data-based system model of a circuit-level amplifier. The circuit-level amplifier is characterized by a .p2d file generated by the system component AmplifierP2D_Setup.
2. A .p2d file contains small- and large-signal 2-port S-parameter data with optional noise parameters and intermodulation table data. This model uses data related to amplifier modeling only; it ignores frequency translation and intermodulation table data.
3. For transient analysis, only the small signal S-parameters are used; these are specified in the top portion of the ACDATA block of the .p2d file.
4. For large-signal frequency analysis (HB, LSSP), the power-dependent S-parameters from the ACDATA block are used. If the input signal power or frequency is out of range, data in the ACDATA block is extrapolated beyond the minimum or maximum value in the ACDATA block, depending on which boundary was exceeded.
5. This model blocks dc.
6. For circuit envelope simulation, baseband signals are blocked.
7. AmplifierP2D is implemented using the FDD model. Unlike Amplifier and AmplifierS2D, AmplifierP2D does not produce harmonics or intermods in a Harmonic Balance simulation.

In a Harmonic Balance analysis, the only frequency component that can pass through AmplifierP2D is that specified by Freq. Signals at all other frequencies will see a ground at the input of Amplifier P2D. Similarly, in a Circuit Envelope simulation, the envelope signal around the carrier frequency Freq will pass

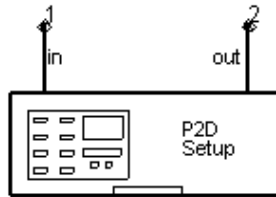
through AmplifierP2D and get distorted. The envelope signals around all other carrier frequencies are ignored.

If you need any kind of multi-tone behavior, consider using Amplifier S2D instead.

8. For Harmonic Balance, Circuit Envelope, or LSSP simulation, setting the Freq parameter to the pre-defined variable *freq* causes simulation errors. For these simulations, Freq must be set to the fundamental frequency, which can be accessed through the pre-defined variable *_freq1*.
9. For the large-signal S-parameter (.p2d) data format, refer to the Circuit Simulation manual, Chapter 6, Working with Data Files, the section, “P2D Format.”
10. An example of the use of this device is provided; access the example from the ADS Main window: *File > Example Project > BehavioralModels > AmplifierS2DandP2D_prj*.

AmplifierP2D_Setup (P2D File Amplifier; FDD-Based, for Single Carrier Signal Setup)

Symbol



Parameters

Filename = File name for P2D file, unitless

Order = Order for Harmonic Balance analysis, integer

Freq_Start = Frequency start value, in Hz

Freq_Stop = Frequency stop value, in Hz

Freq_Step = Frequency step value, in Hz

Pin_Start = Input power start value, in dBm

Pin_Stop = Input power stop value, in dBm

Pin_Step = Input power step value, in dB

Range of Usage

Order > 0

Freq_Start \geq 0

Freq_Stop \geq 0

Freq_Step \geq 0

Notes/Equations

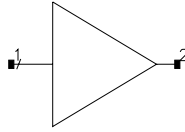
1. AmplifierP2D_Setup performs a swept input power and swept frequency P2D (Harmonic Balance) simulation of a circuit-level amplifier and generates a P2D file for subsequent use by the data-based system model AmplifierP2D.
2. AmplifierP2D_Setup can be pushed into for a view of the implementation. If necessary, the component can be copied and modified to suit individual needs.
3. In order for AmplifierP2D to produce an accurate model of the circuit-level amplifier characterized via AmplifierP2D_Setup, the Order parameter for

AmplifierP2D_Setup must be large enough to prevent aliasing of higher-order frequency components. For mildly non-linear circuits, an Order of around five should suffice while an Order in the range 10-15 is recommended for highly non-linear circuits.

4. An example of the use of this device is provided; access the example from the ADS Main window: *File > Example Project > BehavioralModels > AmplifierS2DandP2D_prj.*

AmplifierS2D (S2D File Amplifier, Polynomial Model for Nonlinearity)

Symbols:



Parameters

S2DFile = Filename for S2D data

SSfreq = Small-signal frequency for interpolating S-parameters: auto (default setting, implies _freq1 for _hb_state = 1 (HB simulations) and freq otherwise), freq, _freq1, _freq2, _freq3. Best results for multitone HB simulations are obtained by using the freq setting.

InterpMode = Interpolation mode

InterpDom = Interpolation data

GCFreq = Reference frequency for gain compression

VarName = Variable that parameterizes S2D data

VarValue = Value of variable referenced in S2D file

ImpNoncausalLength = Non-causal function impulse response order

ImpMode = Convolution mode

ImpMaxFreq = Maximum frequency to which device is evaluated

ImpDelta Freq = Sample spacing in frequency

ImpMaxOrder = Maximum allowed impulse response order

ImpWindow = Smoothing window

ImpRelTol = Relative impulse response truncation factor

ImpAbsTol = Absolute impulse response truncation factor

Range of Usage

S2D gain compression valid parameter ranges:

GComp3: $IP3 > 1DBC + 10.6$

GComp4: $IP3 > PS + 8.6$

GComp5: $PS > 1DBC + 3$

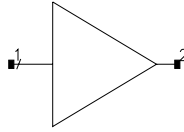
GComp6: $PS > 1DBC + 3, IP3 > 1DBC + 10.6$

Notes/Equations

1. An .s2d file contains small signal S-parameter data with optional noise parameters, gain compression parameters, and intermodulation table data. This model uses data related to amplifier modeling only; frequency translation and intermodulation table data are ignored.
2. When different sets of GComp7 data are given for different frequencies, GCFreq must be specified for Amplifier S2D to use a particular set of data. Similarly, when multiple sets of GComp7 data are given for a swept variable, VarName and VarValue must be specified for Amplifier S2D to use a particular set of data.
3. The .s2d file contains small-signal S-parameter data blocks, noise data blocks, and GCOMP1 - GCOMP7 nonlinearity data blocks for modeling amplifiers and mixers. Only odd order harmonics are modeled in harmonic balance simulation for GCOMP i ($i = 1-7$) nonlinearity. Even order harmonics are not included because this information cannot be derived from the GCOMP i specification.
4. This model blocks dc.
5. For circuit envelope simulation, baseband signals are blocked.
6. For the small- or large-signal S-parameter (.s2d) data format, refer to the Circuit Simulation manual, Chapter 6, Working with Data Files, in the section, "S2D Format."
7. An example of the use of this device is provided; access the example from the ADS Main window: *File > Example Project > BehavioralModels > AmplifierS2DandP2D_prj.*

AmpLoadPull (SDD Load-Pull Amplifier)

Symbol



Parameters

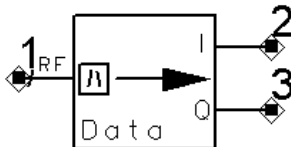
Dataset = Name of dataset containing load-pull data for this amplifier model (can be file-based)

Notes/Equations

1. For amplifier designers, the AmpLoadPull and LoadPullSetup components address the issue of output match in one or more frequency bands. This is often investigated via load-pull contours indicating the load impedances that, when presented to the output of an amplifier with a given source impedance and power, cause a certain power to be delivered to the load. The LoadPullSetup component extracts a dataset, given ranges and steps for input power, output reflection coefficient magnitude and output reflection coefficient angle. AmpLoadPull uses this dataset and allows fast behavioral amplifier simulations for all input power and output reflection coefficient values in the specified ranges.
2. This amplifier model uses the load-pull data generated by the LoadPullSetup component. It is useful in modeling nonlinear amplifiers with or without input/output matching networks. This model can be used in all simulations.
3. When using this model in other simulations, please make sure the input power and the load value are within the range of the original data.
4. An example of the use of this device is provided; access the example from the ADS Main window: *File > Example Project > BehavioralModels > AmpLoadPull_prj.*

IQ_Demod_Data (IQ Demodulator Behavioral Model)

Symbol



Parameters

Dataset = Name of dataset name generated by IQ_Demod_Setup, unitless

Freq = Modulation carrier frequency, in Hz

Range of Usage

Freq ≥ 0

Notes/Equations

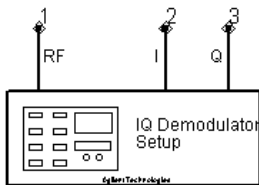
1. IQ_Demod_Data is a data-based system model of a circuit-level I/Q demodulator. The circuit-level demodulator is characterized by a dataset generated by the system component IQ_Demod_Setup.
2. The demodulation behaviors are extracted for a single carrier frequency. Only the demodulation distortions are characterized, so it is possible to set the frequency parameter in IQ_Demod_Data to differ from the value set in IQ_Demod_Setup. The demodulation will occur at the frequency set in IQ_Demod_Data. The extracted distortions to the baseband signal as it passes through IQ_Demod_Data will be applied at this frequency. IQ_Demod_Data does not model any dispersion in these distortions across the baseband bandwidth.
3. IQ_Demod_Data is designed for Circuit Envelope system verification where various filters typically eliminate unwanted frequency components. Therefore, IQ_Demod_Data makes no attempt to model harmonic components. The signals generated on the I and Q output ports are purely baseband signals. IQ_Demod_Data does not generate any frequency components at any carrier frequency.
4. In order for IQ_Demod_Data to produce an accurate model of the circuit-level demodulator characterized via IQ_Demod_Setup, the IQ_Demod_Setup Order parameter must be large enough to prevent aliasing of higher-order frequency components. For mildly non-linear circuits, an Order of about 5 should suffice,

while an Order in the range of 10-15 is recommended for highly non-linear circuits. Providing this criterion is met, the Order parameter for the simulation controller can be very low when using IQ_Demod_Data in a Circuit Envelope simulation. Typically, an Order of around three should suffice for accurate demodulation modeling. Note, however, that accurate modeling of other components of the complete system may necessitate a larger value for this parameter.

5. For a circuit-level demodulator, the impedances presented by the input and output pins will generally be a complicated function of all state variables within the demodulator. For the IQ_Demod_Data, a certain simplification is necessary as no information about the environment in which the demodulator will later reside is known at the time of extraction. The input impedance at the RF input pin is extracted at the RF carrier frequency specified in IQ_Demod_Setup, and is a function of the input power. The impedances at the I and Q output pins are extracted at DC, and are modeled as constant impedances.
6. The extrapolation properties of IQ_Demod_Data above Pin_Stop are very poor. When using IQ_Demod_Data, make sure that the RF input signal does not exceed the Pin_Stop value set in IQ_Demod_Setup when the model was extracted. Extrapolation to signal levels below the Pin_Start value set in IQ_Demod_Setup will generally be good as long as the Pin_Start value lies within the linear operating range of the demodulator.
7. IQ_Demod_Data does not model noise.
8. An example of the use of this device is provided; access the example from the ADS Main window: *File > Example Project > BehavioralModels > IQ_Demod_prj.*

IQ_Demod_Setup (IQ Demodulator Setup)

Symbol



Parameters

Freq = Modulation carrier frequency, in Hz

Order = Order for Harmonic Balance analysis, integer

Pin_Start = Modulation power sweep start value, in dBm referenced to a 50 Ohm system

Pin_Stop = Modulation power sweep stop value, in dBm referenced to 50 Ohm system

Pin_Step = Modulation power sweep step value, in dB

Range of Usage

Freq ≥ 0

Order > 0

Pin_Stop $\geq 10 + 20 \log_{10}(v_{\max})$

Where v_{\max} is the maximum peak voltage magnitude of the modulated signal.

Pin_Stop $\geq P_{\max,z} + 10 \log_{10}(z/50)$

Where $P_{\max,z}$ is the maximum modulated signal power in dBm for a z Ohm system.

Notes/Equations

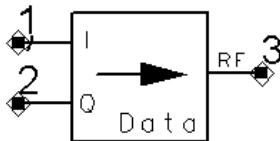
1. IQ_Demod_Setup performs a swept Harmonic Balance simulation of a circuit-level I/Q demodulator and generates a dataset for subsequent use by the data-based system model IQ_Demod_Data.
2. In order for IQ_Demod_Data to produce an accurate model of the circuit-level demodulator characterized by IQ_Demod_Setup, the Order parameter for IQ_Demod_Setup must be large enough to prevent aliasing of higher-order frequency components. For mildly non-linear circuits, an Order of around five

should suffice, while an Order in the range of 10-15 is recommended for highly non-linear circuits.

3. The demodulation behaviors are extracted for a single carrier frequency and over the range of modulated signal power specified by Pin_Start and Pin_Stop. These power values are specified in dBm with respect to a 50 Ohm system. For an accurate IQ_Demod_Data behavioral model, Pin_Stop must be set at least as large as the expected maximum modulated signal power level. See the Range of Usage section for conversion if working in other than a 50 Ohm system.
4. IQ_Demod_Setup can be pushed into, for a view of the swept Harmonic Balance simulation controllers. If necessary, the component can be copied and modified to suit individual needs. Typically, this would entail changing the Harmonic Balance controller “HB1” in order to achieve more efficient simulations. See *Circuit Simulation>Chapter 7, Harmonic Balance Simulation* for further details.
5. An example of the use of this device is provided; access the example from the ADS Main window: *File > Example Project > BehavioralModels > IQ_Demod_prj*.

IQ_Mod_Data (IQ Modulator Behavioral Model)

Symbol



Parameters

Dataset = Name of dataset name generated by IQ_Mod_Setup, unitless

Freq = Modulation carrier frequency, in Hz

Range of Usage

Freq ≥ 0

Notes/Equations

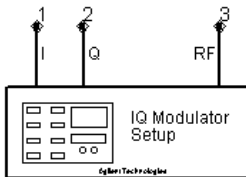
1. IQ_Mod_Data is a data-based system model of a circuit-level I/Q modulator. The circuit-level modulator is characterized by a dataset generated by the system component IQ_Mod_Setup.
2. The modulation behaviors are extracted for a single carrier frequency. Only the modulation distortions are characterized, so it is possible to set the frequency parameter in IQ_Mod_Data to differ from the value set in IQ_Mod_Setup. The modulation will occur at the frequency set in IQ_Mod_Data. The extracted distortions to the baseband signal as it passes through IQ_Mod_Data will be applied at this frequency. IQ_Mod_Data does not model any dispersion in these distortions across the baseband bandwidth.
3. IQ_Mod_Data is designed for Circuit Envelope system verification where various filters typically eliminate unwanted frequency components. Therefore, IQ_Demod_Data makes no attempt to model harmonic components. The signal generated on RF output port is only a baseband signal modulated onto the carrier frequency. IQ_Mod_Data does not generate any frequency components at DC or at any harmonic of the carrier frequency.
4. In order for IQ_Mod_Data to produce an accurate model of the circuit-level demodulator characterized via IQ_Mod_Setup, the Order parameter for IQ_Mod_Setup must be large enough to prevent aliasing of higher-order frequency components. For mildly non-linear circuits, an Order of around five should suffice, while an Order in the range of 10-15 is recommended for highly

non-linear circuits. Providing this criterion is met, the Order parameter for the simulation controller can be very low when using IQ_Mod_Data in a Circuit Envelope simulation. Typically, an Order of around three should suffice for accurate demodulation modeling. Note, however, that accurate modeling of other components of the complete system may necessitate a larger value for this parameter.

5. For a circuit-level modulator, the impedances presented by the input and output pins will generally be a complicated function of all state variables within the modulator. For the IQ_Mod_Data, a certain simplification is necessary as no information about the environment in which the modulator will later reside is known at the time of extraction. Specifically, IQ_Mod_Data presents a constant inward-looking impedance at the input and output pins. The input impedances at the baseband input pins are extracted at DC, while the impedance at the RF output pin is extracted at the RF carrier frequency specified in IQ_Mod_Setup.
6. The extrapolation properties of IQ_Mod_Data above Pin-Stop are very poor. When using IQ_Mod_Data, please make sure that the input baseband signal does not exceed the Pin_Stop value set in IQ_Mod_Setup when the model was extracted. Note that the Pin_Stop value is in reference to the complex baseband signal, which is typically a factor of $\sqrt{2}$, or 3 dB, larger than the power of the signal at either individual input pin. Extrapolation to signal levels below the Pin_Start value set in IQ_Mod_Setup will generally be good as long as the Pin_Start value lies within the linear operating range of the modulator.
7. IQ_Mod_Data does not model noise.
8. An example of the use of this device is provided; access the example from the ADS Main window: *File > Example Project > BehavioralModels > IQ_Mod_prj*.

IQ_Mod_Setup (IQ Modulator Setup)

Symbol



Parameters

Freq = Modulation carrier frequency, in Hz

Order = Order for Harmonic Balance analysis, integer

Pin_Start = Modulation power sweep start value, in dBm referenced to a 50 Ohm system

Pin_Stop = Modulation power sweep stop value, in dBm referenced a 50 Ohm system

Pin_Step = Modulation power sweep step value, in dB

Range of Usage

Freq ≥ 0

Order > 0

Pin_Stop $\geq 10 + 20 \log_{10}(v_{\max})$

where v_{\max} is the maximum peak voltage magnitude of the modulation signal

$$[\sqrt{|V_i|^2 + |V_q|^2}]$$

Pin_Stop $\geq P_{\max,z} + 10 \log_{10}(z/50)$

where $P_{\max,z}$ is the maximum modulation signal power in dBm for a z Ohm system.

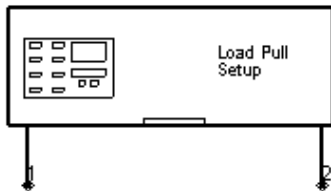
Notes/Equations

1. IQ_Mod_Setup performs a swept Harmonic Balance simulation of a circuit-level I/Q modulator and generates a dataset for subsequent use by the data-based system model IQ_Mod_Data.
2. In order for IQ_Mod_Data to produce an accurate model of the circuit-level demodulator characterized by IQ_Mod_Setup, the Order parameter for IQ_Mod_Setup must be large enough to prevent aliasing of higher-order frequency components. For mildly non-linear circuits, an Order of around five should suffice, while an Order in the range of 10-15 is recommended for highly non-linear circuits.

3. The modulation behaviors are extracted for a single carrier frequency and over the range of modulation signal power specified by `Pin_Start` and `Pin_Stop`. These power values are specified in dBm with respect to a 50 Ohm system. For an accurate `IQ_Mod_Data` behavioral model, `Pin_Stop` must be set at least as large as the expected maximum modulated signal power level. See the Range of Usage section for conversion if working in other than a 50 Ohm system.
4. `IQ_Mod_Setup` can be pushed into for a view of the swept Harmonic Balance simulation controllers. If necessary, the component can be copied and modified to suit individual needs. Typically, this would only entail changing the Harmonic Balance controller “HB1” in order to achieve more efficient simulations. See *Circuit Simulation > Chapter 7, Harmonic Balance Simulation* for further details.
5. An example of the use of this device is provided; access the example from the ADS Main window: *File > Example Project > BehavioralModels > IQ_Mod_prj*.

LoadPullSetup (Load Pull Setup)

Symbol



Parameters

Freq = Frequency in Hz, kHz, Mhz, or GHz (default: 1 GHz)

Order = Order for harmonic balance

Pin_Start = Input power start value

Pin_Stop = Input power stop value

Pin_Step = Input power step value

GamAng_Start = Output reflection coefficient angle start value

GamAng_Stop = Output reflection coefficient angle stop value

GamAng_Step = Output reflection coefficient angle step value

GamMag_Start = Output reflection coefficient magnitude start value

GamMag_Stop = Output reflection coefficient magnitude stop value

GamMag_Step = Output reflection coefficient magnitude step value

Range of Usage

Freq ≥ 0

Order > 0

Notes/Equations

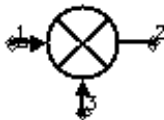
1. For amplifier designers, the AmpLoadPull and LoadPullSetup components address the issue of output match in one or more frequency bands. This is often investigated via load-pull contours indicating the load impedances that, when presented to the output of an amplifier with a given source impedance and power, cause a certain power to be delivered to the load. The LoadPullSetup component extracts a dataset, given ranges and steps for input power, output reflection coefficient magnitude and output reflection coefficient angle.

AmpLoadPull uses this dataset and allows fast behavioral amplifier simulations for all input power and output reflection coefficient values in the specified ranges.

2. Although the behavioral data collected via LoadPullSetup is found using Harmonic Balance simulations, the subsequent behavioral amplifier simulations are not restricted to Harmonic Balance. In fact, the behavioral amplifier is assumed to be used for subsequent system verification (BER, ACPR etc. for Tx, Rx) within a Circuit Envelope framework. In such applications, various filters typically eliminate unwanted DC and harmonic components. Therefore, the load-pull suite was not designed to predict such components. This means that
 - for a 1-tone Harmonic Balance simulation at frequency f , only the frequency component at f (fundamental) is retained. All others will be invalid or zero as they are assumed to be filtered later.
 - for a 2-tone Harmonic Balance simulation at frequencies f_1 and f_2 (practically very close), only the frequency components at f_1 and f_2 (fundamentals) and those at $2*f_1-f_2$ and $2*f_2-f_1$ (intermodulation) are retained as these will be very close and cannot be assumed to be filtered. All others will be invalid or zero as they are assumed to be filtered later.
3. LoadPullSetup can be pushed into for a view of the implementation. If necessary, the component can be copied and modified to suit individual needs.
4. An example of the use of this device is provided; access the example from the ADS Main window: *File > Example Project > BehavioralModels > AmpLoadPull_prj.*

MixerHBdata (2-Tone HB Mixer)

Symbol



Parameters

Dataset = Dataset name for two-tone Harmonic Balance mixer (can be file-based).

RFindexExpr = The RF tone mixing index defined in the MixerHBsetup component. Typically set to "Mix(2)" and not edited by the user.

LOindexExpr = The LO tone mixing index defined in the MixerHBsetup component. Typically set to "Mix(1)" and not edited by the user.

IFwaveExpr = The IF wave variable defined in the MixerHBsetup component. Typically set to "b2" and not edited by the user.

IMdata = Intermodulation data flag. Set to -1 or to 1 for selecting lower or upper sideband respectively from dataset for mixer modeling. For additional information refer to Note 4.

ConvGain = Conversion gain multiplier from RF port (Port 1) to IF port (Port 2). Use $x+j*y$, polar (x, y), dbpolar (x, y) for complex value.

SP11 = Denotes RF port reflection. Use $x+j*y$, polar (x, y), dbpolar (x, y), vswrpolar (x,y) for complex value.

SP12 = Denotes IF port to RF port leakage. Use $x+j*y$, polar (x, y), dbpolar (x, y), vswrpolar (x,y) for complex value.

SP13 = Denotes LO port to RF port leakage. Use $x+j*y$, polar (x, y), dbpolar (x, y), vswrpolar (x,y) for complex value.

SP21 = Denotes RF port to IF port leakage. Use $x+j*y$, polar (x, y), dbpolar (x, y), vswrpolar (x,y) for complex value.

SP22 = Denotes IF port reflection. Use $x+j*y$, polar (x, y), dbpolar (x, y), vswrpolar (x,y) for complex value.

SP23 = Denotes LO port to IF port leakage. Use $x+j*y$, polar (x, y), dbpolar (x, y), vswrpolar (x,y) for complex value.

SP31 = Denotes RF port to LO port leakage. Use $x+j*y$, polar (x, y), dbpolar (x, y), vswrpolar (x,y) for complex value.

SP32 = Denotes IF port to LO port leakage. Use $x+j*y$, polar (x, y), dbpolar (x, y), vswrpolar (x,y) for complex value.

SP33 = Denotes LO port reflection. Use $x+j*y$, polar (x, y), dbpolar (x, y), vswrpolar (x,y) for complex value.

NF = Input double sideband noise figure, in dB

NFmin = Minimum double sideband noise figure at S_{opt} , in dB

S_{opt} = Optimum source reflection for minimum noise figure. Use $x+j*y$, polar (x, y), dbpolar (x, y), vswrpolar (x,y) for complex value

R_n = Equivalent noise resistance, in mOhm, Ohm, kOhm, MOhm, or GOhm.

Z_1 = Reference impedance for RF port in mOhm, Ohm, kOhm, MOhm, or GOhm.

Z_2 = Reference impedance for IF port in mOhm, Ohm, kOhm, MOhm, or GOhm.

Z_3 = Reference impedance for LO port in mOhm, Ohm, kOhm, MOhm, or GOhm.

P_{RFnom} = RF power, in dBm

P_{LOnom} = LO power, in dBm

M_{RF} = Number of RF harmonics to be considered from dataset

N_{LO} = Number of LO harmonics to be considered from dataset

Notes/Equations

1. MixerHBdata is a data-based system model of a circuit-level mixer. The dataset used by MixerHBdata is generated from the circuit-level mixer by the behavioral extractor component MixerHBsetup available in the System-Data Models palette. Ports 1, 2 and 3 on MixerHBdata correspond to the RF, IF and LO ports of the MixerHBsetup component. An example using these two MixerHB devices is accessible from the ADS Main window: *File > Example Project > BehavioralModels > MixerHBdata_prj*.
2. MixerHBdata can be used in Harmonic Balance, Circuit Envelope, and Transient simulations.
3. MixerHBdata is currently configured to produce IF output with zero phase regardless of the IF output phase represented in the MixerHBsetup extracted dataset.

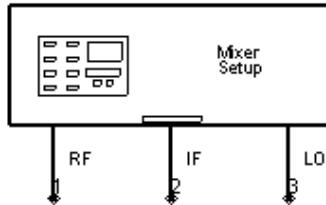
4. A dataset extracted using the MixerHBsetup component contains both upper and lower sideband harmonics and intermodulation products. By formal definition the upper sideband contains frequencies above the higher of the two mixing frequencies. For instance, in a typical RFIC receiver, the center frequency is the RFfreq because LOfreq < RFfreq. In the MixerHBdata component the IMdata parameter is used to prompt the use of only one of these two bands for deriving an internal polynomial model for frequency response. Once this model has been constructed, it is used to represent behavior of both sidebands in the MixerHBdata simulation output.

If a 2-tone Harmonic Balance simulation is to be performed on a MixerHBdata component using Freq[1]=LOfreq and Freq[2]=RFfreq in the simulator component, then setting IMdata=-1 selects the lower sideband frequencies and IMdata=1 selects the upper sideband frequencies from the dataset. The harmonics and intermodulation products are defined as $|m*RFfreq - n*LOfreq|$ in the lower sideband and $|m*RFfreq + n*LOfreq|$ in the upper sideband, where m and n are integers bounded by M_RF and N_LO (the maximum harmonic order parameters of the MixerHBdata component).

By the convention that $-M_RF \leq m \leq M_RF$ and $-N_LO \leq n \leq N_LO$, and the assumption that M_RF and N_LO are of comparable magnitudes, the difference frequencies $|m*RFfreq - n*LOfreq|$ are guaranteed to be lower than RFfreq and the summed frequencies $|m*RFfreq + n*LOfreq|$ are guaranteed to be higher than RFfreq. Thus, the IMdata parameter helps in sideband selection and should be assigned values of -1 or 1 only.

MixerHBsetup (2-Tone HB Mixer Setup)

Symbol



Parameters

RFfreq = RF frequency in Hz, kHz, or MHz, or GHz

LOfreq = LO frequency in Hz, kHz, or MHz, or GHz

RFpwr = RF power, in dBm

LOpwr = LO power, in dBm

RForder = Number of RF harmonics

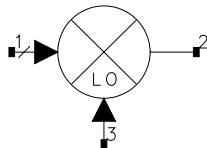
LOorder = Number of LO harmonics

Notes/Equations

1. MixerHBsetup extracts the behavior of a circuit-level mixer using a Harmonic Balance simulation. The extracted dataset can be subsequently used by the corresponding data-based system model MixerHBdata. Ports RF, IF and LO ports of the MixerHBsetup component correspond to the ports 1, 2 and 3 on the MixerHBdata component. An example using these two MixerHB devices is accessible from the ADS Main window: *File > Example Project > BehavioralModels > MixerHBdata_prj*.
2. MixerHBsetup can be pushed into for a view of the extractor implementation including access to the built-in Harmonic Balance simulator component. Note that equations and variables of the extractor subcircuit are referenced by name from parameters of the MixerHBdata component. Any user-initiated changes made to a specific instance of MixerHBsetup should be reflected in a complementary instance of MixerHBdata for correct dataset indexing for behavioral simulation.

MixerIMT (Obsolete Intermodulation Table Mixer)

Symbols:



Parameters

SS_SideBand = produce UPPER or LOWER sideband at output port for linear analysis (ignored for nonlinear analysis—for example, Harmonic Balance)

ConvGain = conversion gain (real or complex number; see note 5)

S11 = port 1 reflection (real or complex number; see note 5)

S22 = port 2 reflection (real or complex number; see note 5)

S33 = port 3 reflection (real or complex number; see note 5)

NF = input double sideband noise figure, in dB

NFmin = minimum double sideband noise figure at Sopt, in dB

Sopt = optimum source reflection for NFmin

Rn = equivalent noise resistance, in ohms

R1 = reference resistance for port 1 (real number)

R2 = reference resistance for port 2 (real number)

R3 = Reference resistance for port 3 (real number)

InThresh = RF input threshold, in volts

LoThresh = LO input threshold, in volts

IMT_File = file containing intermodulation table; accepted formats: IMT, P2D, and S2D

Range of Usage

$NF \geq 0$ dB

$NF_{min} > 0$

$0 < |Sopt| < 1$

$0 < Rn$

InThresh > 0
 LoThresh > 0
 |ConvGain| > 0

Notes/Equations:

1. This component is included for the convenience of customers who have existing designs with the MixerIMT model. New instances can be accessed by typing MixerIMT into the Component History field in the Schematic window, pressing Enter, and moving your cursor to the drawing area to place it.
2. MixerIMT does not function in Transient simulations; MixerIMT blocks baseband signals in Circuit Envelope simulations; MixerIMT does not produce intermodulation among RF input tones (see Note 9). Replace MixerIMT with MixerIMT2 for these functions.
3. S12=0.
4. If no IMT file is referenced, the output (IF) signal is the product of the input RF and LO spectral tones only.
5. If NFmin, Sopt, and Rn are used to characterize noise, the following relation must be satisfied for a realistic model.

$$\frac{Rn}{Zo} \geq \frac{To(Fmin-1)|1 + Sopt|^2}{T^4} \frac{(1 - |S_{11}|^2)}{|1 - Sopt S_{11}|^2}$$

Rn will be reset to a value that meets this criteria if it does not satisfy this condition.

6. Use the function polar(mag,ang), or dbpolar(dB,ang), or VSWRpolar(VSWR, ang) to convert these specifications into a complex number.
7. For an S-parameter or a noise figure sinusoidal ripple, use the function ripple(mag, intercept, period, variable); for example ripple(0-1, 0-1, 10 MHz, Freq).
 Example: S21=dbpolar(10+ripple(0),0.)
8. This model blocks dc.
9. MixerIMT produces intermodulation products at frequencies

$$|m*LO +/- n*Signal|$$

where

$$-M \leq m \leq M \text{ and } -N \leq n \leq N$$

Also, M and N are the orders for LO and RF signal frequencies set in the Harmonic Balance or Circuit Envelope simulation controllers.

If there are multiple frequencies at the RF and LO ports, there will be a superposition of the RF-LO intermodulation tones. For example, given the frequencies LO1 and LO2 at the LO port, and Signal1 and Signal2 at the RF port, there will be intermodulation products at the frequencies

$$|m*LO_1 +/- n*Signal_1|, |m*LO_1 +/- n*Signal_2|,$$

$$|m*LO_2 +/- n*Signal_1|, |m*LO_2 +/- n*Signal_2|$$

where

$$-M_{RF} \leq m \leq M_{RF} \text{ and } -N_{LO} \leq n \leq N_{LO}$$

There will be no products at $|LO_1 +/- LO_2|$, $|Signal_1 +/- Signal_2|$ or any other RF or LO self-modulation products.

Intermodulation Product Table File Used with MixerIMT

The intermodulation product table (.imt) file is a user-defined table of mixer intermodulation (IM) products between the LO and input signal. It relates your mixer IM output level to the output signal level. The output signal is a direct mapping of each input tone with each LO tone. Interaction between input signal tones are not characterized in this model.

Two example IMT files are provided: *dbl1.imt* (Figure 8-1) and *dbl2.imt* (Figure 8-2). After selecting the IMT_File parameter, click on the *Copy template* button in the dialog box to select a file.

```

BEGIN IMT_DATA
! DLB1.IMT
! Intermodulation table for double balanced mixer #1
! Signal Level (dBm)   Lo Level (dBm)
# IMT   (-10      7)
!      M x Lo ( Horizontal )   N x Signal (Vertical )
% 0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15
!
99  26  35  39  50  41  53  49  51  45  65  55  75  65  85  99 !row0
24   0  35  13  40  24  45  28  49  35  55  45  65  55  99   !row1
73  73  74  70  71  64  69  64  69  65  75  75  85  99   !row2
67  64  69  50  77  47  74  44  74  45  75  55  99   !row3
86  90  86  88  88  85  86  85  90  85  85  99   !row4
90  80  90  71  90  68  90  65  88  65  99   !row5
90  90  90  90  90  90  90  90  90  99   !row6
90  90  90  90  90  87  90  90  99   !row7
99  95  99  95  99  95  99  99   !row8
90  95  90  95  90  99  99   !row9
99  99  99  99  99  99   !row10
90  99  99  99  99   !row11
99  99  99  99   !row12
99  99  99   !row13
99  99   !row14
99   !row15
END

```

Figure 8-1. dbl1.imt File Example

```

BEGIN IMT_DATA
! DLB2.IMT
! Intermodulation table for double balanced mixer #2
! Signal Level (dBm)   Lo Level (dBm)
# IMT      (-10      17)
!   M x Lo ( Horizontal )   N x Signal ( Vertical )
%  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
!
  99  39  42  46  58  37  65  49  75  62  72  61  70  57  87  60  !row0
  25  0   39  11  50  16  59  19  59  43  63  52  70  57  73  99  !row1
  68  67  76  67  80  66  82  66  83  72  84  72  82  70  99  !row2
  63  58  65  60  65  55  64  54  66  57  85  54  70  99  !row3
  96  80  96  80  95  82  98  78  90  95  95  95  99  !row4
  93  73  87  72  88  66  85  64  82  75  95  99  !row5
  99  99  99  99  99  99  99  99  99  99  99  !row6
  99  79  99  78  99  78  99  81  99  99  !row7
  99  95  99  95  99  95  99  95  99  !row8
  90  95  90  90  90  99  90  99  !row9
  99  99  99  99  99  99  99  !row10
  90  99  90  95  90  99  !row11
  99  99  99  99  99  !row12
  90  99  90  99  !row13
  99  99  99  !row14
  99  99  !row15
END

```

Figure 8-2. dbl2.imt File Example

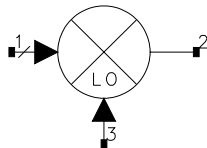
In the IMT table:

- The vertical row number N (0, 1, to 15) indicates the harmonic of the signal used in deriving the spurious output signal.
- The horizontal column number M (0, 1, to 15) indicates the harmonic of the local oscillator used in deriving the spurious output signal.
- In row 2, column 4, the data is 13. This means that for an input signal at -10 dBm input, with an LO drive of $+7$ dBm, an output spurious signal will occur at $3 * LO +/- 1 * signal$, with a level that is 13 dB below the fundamental output signal.
- If the input signal differs from the -10 dBm reference power level listed at the top of the table by X dB, then the number in the table is adjusted by adding $(N-1) \times X$ dB to it. This manner of adjustment is good for input power levels up to 5 dB greater than the reference signal power.

- If the local oscillator signal differs from the +7 dBm reference power level listed at the top of the table by X dB, then the number in the table is adjusted by adding it by $M \times X$ dB to it. This manner of adjustment is good for local oscillator power levels from the reference level minus 10 dB to the reference level plus 3 dB.
- If items are missing from the IMT table, a triangular table, they are loaded as 200 dB down from input reference.

MixerIMT2 (Intermodulation Table Mixer)

Symbols:



Parameters

ConvGain = Conversion gain (real or complex number)

SP11 = RF port 1 reflection; use $x+j*y$, polar (x, y), dbpolar (x, y), vswrpolar (x,y) for complex value

SP12 = IF port to RF port leakage; use $x+j*y$, polar (x, y), dbpolar (x, y), vswrpolar (x,y) for complex value

SP13 = LO port to RF port leakage; use $x+j*y$, polar (x, y), dbpolar (x, y), vswrpolar (x,y) for complex value

SP21 = RF port to IF port leakage; use $x+j*y$, polar (x, y), dbpolar (x, y), vswrpolar (x,y) for complex value

SP22 = IF port reflection; use $x+j*y$, polar (x, y), dbpolar (x, y), vswrpolar (x,y) for complex value

SP23 = LO port to IF port leakage; use $x+j*y$, polar (x, y), dbpolar (x, y), vswrpolar (x,y) for complex value

SP31 = RF port to LO port leakage; use $x+j*y$, polar (x, y), dbpolar (x, y), vswrpolar (x,y) for complex value

SP32 = IF port to LO port leakage; use $x+j*y$, polar (x, y), dbpolar (x, y), vswrpolar (x,y) for complex value

SP33 = LO port reflection; use $x+j*y$, polar (x, y), dbpolar (x, y), vswrpolar (x,y) for complex value

NF = input double sideband noise figure, in dB

NFmin = minimum double sideband noise figure at S_{opt} , in dB

S_{opt} = optimum source reflection for minimum noise figure; use $x+j*y$, polar (x, y), dbpolar (x, y), vswrpolar (x,y) for complex value

Rn = equivalent noise resistance

R1 = reference resistance for port 1 (real number)

R2 = reference resistance for port 2 (real number)

R3 = reference resistance for port 3 (real number)

M_RF = IMT order for RF port

N_LO = IMT order for LO port

IMTValueType = IMT value type (dB_value or dBm_value)

IMT_File = file containing intermodulation table; accepted formats: IMT, P2D, and S2D

Range of Usage

NF ≥ 0 dB

NFmin > 0

0 < | Sopt | < 1

0 < Rn

M_RF, N_LO ≤ 15

Notes/Equations

1. If NFmin, Sopt, and Rn are used to characterize noise, the following relation must be satisfied for a realistic model.

$$\frac{Rn}{Zo} \geq \frac{To(Fmin-1)|1+Sopt|^2}{T^4} \frac{(1-|S_{11}|^2)}{|1-Sopt S_{11}|^2}$$

If the noise parameters attempt to describe a system that requires negative noise (due to Rn being too small), the negative part of the noise will be set to 0 and a warning message will be issued.

2. Use the function polar(mag,ang), or dbpolar(dB,ang), or vswrpolar(VSWR, ang) to convert these specifications into a complex number.
3. For an S-parameter or a noise figure sinusoidal ripple, use the function ripple(mag, intercept, period, variable); for example ripple(0-1, 0-1, 10 MHz, Freq).
Example: S21=dbpolar(10+ripple(),0.)
4. MixerIMT2 produces intermodulation products at frequencies

$$|m*LO +/- n*Signal|$$

where

$$-M_RF \leq m \leq M_RF \text{ and } -N_LO \leq n \leq N_LO$$

If there are multiple frequencies at the RF and LO ports, there will be a superposition of the RF-LO intermodulation tones. For example, given the frequencies LO1 and LO2 at the LO port, and Signal1 and Signal2 at the RF port, there will be intermodulation products at the frequencies

$$|m*LO_1 +/- n*Signal_1|, |m*LO_1 +/- n*Signal_2|,$$

$$|m*LO_2 +/- n*Signal_1|, |m*LO_2 +/- n*Signal_2|$$

where

$$-M_RF \leq m \leq M_RF \text{ and } -N_LO \leq n \leq N_LO$$

In addition, there will be no products at $|LO_1 +/- LO_2|$, $|Signal_1 +/- Signal_2|$ and other RF or LO self-modulation products.

5. MixerIMT2 can be used in all simulations except for Frequency Converting AC analysis.
6. MixerIMT2 leakage terms are specified in two separate ways: the IMT file entries, and the SPij parameters. For example, LO to IF leakage is given by the (M=1, N=0) term in the IMT file as well as the SP23 parameter. When both specifications are given, the total leakage is given by the complex sum of the two specifications. It is generally advisable to leave SP23=SP13=0; in this case, the LO to IF and the RF to IF leakages are specified by the IMT file.

Intermodulation Product Table File Used with MixerIMT and MixerIMT2

The intermodulation product table (.imt) file is a user-defined table of mixer intermodulation (IM) products between the LO and input signal. It relates your mixer IM output level to the output signal level. The output signal is a direct mapping of each input tone with each LO tone. Interaction between input signal tones are not characterized in this model.

Two example IMT files are provided: *dbl1.imt* and *dbl2.imt*. After selecting the IMT_File parameter, click on the *Copy template* button in the dialog box to select a file.

```

BEGIN IMT_DATA
! DLB1.IMT
! Intermodulation table for double balanced mixer #1
! Signal Level (dBm)   Lo Level (dBm)
# IMT      (-10      7)
!      M x Lo ( Horizontal )   N x Signal (Vertical )
% 0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15
!
99 26 35 39 50 41 53 49 51 45 65 55 75 65 85 99 !row0
24  0 35 13 40 24 45 28 49 35 55 45 65 55 99   !row1
73 73 74 70 71 64 69 64 69 65 75 75 85 99     !row2
67 64 69 50 77 47 74 44 74 45 75 55 99       !row3
86 90 86 88 88 85 86 85 90 85 85 99          !row4
90 80 90 71 90 68 90 65 88 65 99             !row5
90 90 90 90 90 90 90 90 90 99               !row6
90 90 90 90 90 87 90 90 99                   !row7
99 95 99 95 99 95 99 99                       !row8
90 95 90 95 90 99 99                           !row9
99 99 99 99 99 99                               !row10
90 99 99 99 99                                 !row11
99 99 99 99                                     !row12
99 99 99                                         !row13
99 99                                             !row14
99                                               !row15
END

```

Figure 8-3. dbl1.imt File Example

```

BEGIN IMT_DATA
! DLB2.IMT
! Intermodulation table for double balanced mixer #2
! Signal Level (dBm)   Lo Level (dBm)
# IMT      (-10       17)
!   M x Lo ( Horizontal )   N x Signal ( Vertical )
%  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
!
  99  39  42  46  58  37  65  49  75  62  72  61  70  57  87  60  !row0
  25  0   39  11  50  16  59  19  59  43  63  52  70  57  73  99  !row1
  68  67  76  67  80  66  82  66  83  72  84  72  82  70  99  !row2
  63  58  65  60  65  55  64  54  66  57  85  54  70  99  !row3
  96  80  96  80  95  82  98  78  90  95  95  95  99  !row4
  93  73  87  72  88  66  85  64  82  75  95  99  !row5
  99  99  99  99  99  99  99  99  99  99  99  !row6
  99  79  99  78  99  78  99  81  99  99  !row7
  99  95  99  95  99  95  99  95  99  !row8
  90  95  90  90  90  99  90  99  !row9
  99  99  99  99  99  99  99  !row10
  90  99  90  95  90  99  !row11
  99  99  99  99  99  !row12
  90  99  90  99  !row13
  99  99  99  !row14
  99  99  !row15
END

```

Figure 8-4. dbl2.imt File Example

In the IMT table:

- The vertical row number N (0, 1, to 15) indicates the harmonic of the signal used in deriving the spurious output signal.
- The horizontal column number M (0, 1, to 15) indicates the harmonic of the local oscillator used in deriving the spurious output signal.
- In row 2, column 4, the data is 13. This means that for an input signal at -10 dBm input, with an LO drive of +7dBm, an output spurious signal will occur at $3 * LO +/- 1 * signal$, with a level that is 13 dB below the fundamental output signal.
- If the input signal differs from the -10 dBm reference power level listed at the top of the table by X dB, then the number in the table is adjusted by adding $(N-1) \times X$ dB to it. This manner of adjustment is good for input power levels up to 5 dB greater than the reference signal power.

- If the local oscillator signal differs from the +7 dBm reference power level listed at the top of the table by X dB, then the number in the table is adjusted by adding it by $M \times X$ dB to it. This manner of adjustment is good for local oscillator power levels from the reference level minus 10 dB to the reference level plus 3 dB.
- If items are missing from the IMT table, a triangular table, they are loaded as 200 dB down from the input reference.

Example

This example table is for small-signal input conditions for which the IM level is fixed by the type of mixer and LO drive. It is an example of a characteristic mixer at a specific LO drive level; it does not give IM results for multi-tone signal input. In the example, the IM product for $5 \times LO + 1 \times Signal$ is 24 dB below the fundamental output; the IM product for $2 \times LO + 3 \times Signal$ is 69 dB below the fundamental output.

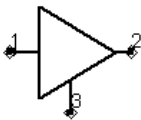
MixerIMT2 also supports an IMT table that contains absolute dBm values at a particular RF level and a particular LO level. The IMT table is not necessarily 15x15. Following is such an example. Please make sure that `IMTValueType=dBm_value` in this case.

In this table, it is shown that for an RF input of -30 dBm and a LO input of 0 dBm, the 1x1 product at output is -29 dBm. The LO feedthrough (0x1 product) is -7.8 dBm. The RF feedthrough (1x0 product) is -30.8 dBm etc.

!	Signal Level (dBm)					LO Level (dBm)									
!	-30					0									
!	M x LO (Horizontal)					N x Signal (Vertical)									
!\	0	1	2	3	4	5	6	7	8	9	10	11	12	13	
14	15														
!															
	-110	-7.8	-35.7	-30.2	-53.2	-38.8	-39.5								
	-30.8	-29	-49.5	-68.2	-71	-86.2	-74.3								
	-106.2	-79.2	-74.5	-92.3	-93	-99.2	-106.8								
	-110	-110	-110	-105.2	-110	-110	-110								
	-110	-110	-110	-110	-110	-110	-110								
	-110	-110	-110	-105.2	-110	-110	-110								
	-110	-110	-110	-110	-110	-110	-110								

VCA_Data (Voltage Controlled Amplifier)

Symbol



Parameters

Dataset = Name of dataset generated by VCA_Setup, unitless

Notes/Equations

1. VCA_Data is a data-based system model of a circuit-level VCA. The circuit-level VCA is characterized by a dataset generated by the system component VCA_Setup.
2. VCA_Data is the most advanced data-based system amplifier. If no special circumstances make AmplifierS2D, AmplifierP2D, AmpH1H2 or AmpLoadPull superior to VCA_Data, the latter model should be used for data-based system modeling of amplifiers.
3. VCA_Setup and VCA_Data work equally well for amplifiers with or without a control voltage. For amplifiers without a control voltage, the control pin should be grounded.
4. VCA_Data is designed for Harmonic Balance or Circuit Envelope system verification (ACPR, BER, etc.) where various filters typically eliminate unwanted frequency components. Therefore, VCA_Data makes no attempt to model harmonic components. This means that for a 1-tone Harmonic Balance simulation at frequency f , only the frequency component at f (fundamental) is retained. All others will be invalid or zero as they are assumed to be filtered later.

For a 2-tone Harmonic Balance simulation at frequencies f_1 and f_2 (practically very close), only the frequency components at f_1 and f_2 (fundamentals) and those at $2 \times f_1 - f_2$ and $2 \times f_2 - f_1$ (intermodulation) are retained as these will be very close and cannot be assumed to be filtered. All others will be invalid or zero as they are assumed to be filtered later. Similarly, for a Circuit Envelope simulation, narrowband modulated components around the carrier(s) will be retained while all others will be invalid or zero as they are assumed to be filtered later.

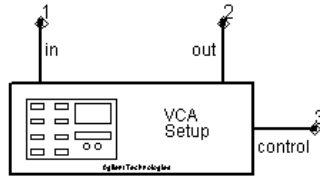
VCA_Data will also run in Transient but this is not a recommended simulation controller. Since VCA_Setup is based on Harmonic Balance analysis, the dataset created by VCA_Setup contains steady-state information only. Transient analysis can predict the correct steady-state response only and will almost certainly result in an incorrect transient response.

5. In order for VCA_Data to produce an accurate model of the circuit-level VCA characterized via VCA_Setup, the Order parameter for VCA_Setup must be large enough to prevent aliasing of higher-order frequency components. For mildly non-linear circuits, an Order around 5 should suffice while a range of 10-15 is recommended for highly non-linear circuits. Provided this criterion is met, the Order parameter for the simulation controller can be very low when using VCA_Data in a Harmonic Balance or Circuit Envelope simulation. Typically, an Order around 3 should suffice for accurate VCA modeling. However, note that accurate modeling of other components of the complete system may necessitate a larger value for this parameter.
6. For a circuit-level VCA, the impedances presented by the input, output and control pins will generally be highly complicated functions of all state variables within the circuit of which the VCA is part. For a data-based VCA, a simplification is necessary as no information about the environment in which the VCA will later reside is known at the time of extraction. Specifically, VCA_Data presents an input power and control voltage dependent impedance at the input, output and control pins, with the input and control impedances assuming an open-circuit output pin. The input and output impedances are voltage-to-current ratios at RF while the control impedance is a voltage-to-current ratio at DC.
7. Assuming proper sampling, the interpolation properties of VCA_Data are generally good but occasionally break down near the limits of the ranges to which input power and control voltage were constrained when generating the dataset using VCA_Setup. To safe-guard against such a breakdown, the upper and lower limits of these ranges can be extended about 6 dB in input power and 1 V in control voltage.
8. The extrapolation properties of VCA_Data are very poor. When using VCA_Data, please make sure input power and control voltage are within the ranges to which input power and control voltage were constrained when generating the dataset using VCA_Setup.
9. VCA_Data does not model noise.

10. VCA_Setup and VCA_Data can be used for simulating an amplifier with an arbitrary swept control parameter, for example a temperature. Simply convert a 2-pin amplifier dependent on a given parameter, in this case Par, to a 3-pin amplifier with the third pin connected to a grounded DC voltage source whose strength equals Par. By sweeping the control voltage of VCA_Setup in the desired parameter range for Par and subsequently connecting the control pin of VCA_Data to a grounded DC voltage source, Par values in the appropriate range can be selected by simply setting the strength of this voltage source properly.
11. An example of the use of this device is provided; access the example from the ADS Main window: *File > Example Project > BehavioralModels > VCA_prj*.

VCA_Setup (Voltage Controlled Amplifier Setup)

Symbol



Parameters

Freq = Frequency, in Hz

Order = Order for Harmonic Balance analysis, integer

Pin_Start = Input power start value, in dBm referenced to a 50 Ohm system

Pin_Stop = Input power stop value, in dBm referenced to a 50 Ohm system

Pin_Step = Input power step value, in dB

Vcontrol_Start = Control voltage start value, in V

Vcontrol_Stop = Control voltage stop value, in V

Vcontrol_Step = Control voltage step value, in V

Range of Usage

Freq ≥ 0

Order > 0

Notes/Equations

1. VCA_Setup performs a swept control voltage and swept input power simulation of a circuit-level VCA and generates a dataset for subsequent use by the data-based system model VCA_Data.
2. VCA_Setup and VCA_Data work equally well for amplifiers with or without a control voltage. For amplifiers without a control voltage, the control pin should be grounded.
3. In order for VCA_Data to produce an accurate model of the circuit-level VCA characterized via VCA_Setup, the Order parameter for VCA_Setup must be large enough to prevent aliasing of higher-order frequency components. For mildly non-linear circuits, an Order of around 5 should suffice while an Order in the range 10-15 is recommended for highly non-linear circuits.

4. VCA_Setup can be pushed into for a view of the swept Harmonic Balance simulation controllers. If necessary, the component can be copied and modified to suit individual needs. Typically, this would only entail changing the Harmonic Balance controller *HBI* in order to achieve more efficient simulations. See *Circuit Simulation > Chapter 7, Harmonic Balance Simulation* for details.
5. An example of the use of this device is provided; access the example from the ADS Main window: *File > Example Project > BehavioralModels > VCA_prj*.
6. If you have a circuit containing VCA_Setup from ADS 2002, you will see a warning message:

SYM_VCA_Setup: Could not find design Design file warning

You must delete the existing VCA_Setup component and insert a new VCA_Setup component.