Agilent Technologies

Advanced Design System 2011.01

Feburary 2011
Timed Components

Acknowledgments

POSSIBILITY OF SUCH DAMAGE.

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

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

UMFPACK 5.0.2 - UMFPACK Notice: The UMFPACK code was modified. Used by permission. UMFPACK Copyright: UMFPACK Copyright © 1995-2006 by Timothy A. Davis. All Rights Reserved. UMFPACK License: Your use or distribution of UMFPACK or any modified version of UMFPACK implies that you agree to this License. This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA Permission is hereby granted to use or copy this

program under the terms of the GNU LGPL, provided that the Copyright, this License, and the Availability of the original version is retained on all copies. User documentation of any code that uses this code or any modified version of this code must cite the Copyright, this License, the Availability note, and "Used by permission." Permission to modify the code and to distribute modified code is granted, provided the Copyright, this License, and the Availability note are retained, and a notice that the code was modified is included. UMFPACK Availability: http://www.cise.ufl.edu/research/sparse/umfpack UMFPACK (including versions 2.2.1 and earlier, in FORTRAN) is available at http://www.cise.ufl.edu/research/sparse . MA38 is available in the Harwell Subroutine Library. This version of UMFPACK includes a modified form of COLAMD Version 2.0, originally released on Jan. 31, 2000, also available at http://www.cise.ufl.edu/research/sparse . COLAMD V2.0 is also incorporated as a built-in function in MATLAB version 6.1, by The MathWorks, Inc. http://www.mathworks.com . COLAMD V1.0 appears as a column-preordering in SuperLU (SuperLU is available at http://www.netlib.org). UMFPACK v4.0 is a built-in routine in MATLAB 6.5. UMFPACK v4.3 is a built-in routine in MATLAB 7.1.

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

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

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

Warranty The material contained in this document is provided "as is", and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Agilent disclaims all warranties, either express or implied,

with regard to this documentation and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Agilent shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Agilent and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

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

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

RF Feedback Loop Example Designs

As typical applications of the RF feedback loops, the RF feedback loop examples created in ADS demonstrate basic phase lock loop and automatic gain control loop designs. These designs are intended for RF feedback subsystem designers to get an idea of what nominal system performance would be. Evaluations can be made regarding degraded system performance due to system impairments that may include nonideal component performance.

These designs can be accessed from the ADS Main window: *File* > *Open* > *Example* > *Timed* > *RF_PLL_Examples_wrk* or *RF_AGC_Loop_wrk*. The designs use the new *CktAGCLoopFilter* (timed) and *RF_DemodExtOscRC* (timed) (Timed Library), and *AverageCxWOffset* (numeric), *DelayEstimator* (numeric), and *DSampleWOffset* (numeric) (Numeric Library), components, as well as existing basic ADS components.

Phase Lock Loop Designs

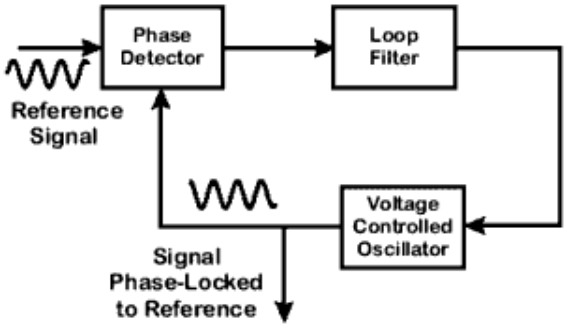
Previously, receiver designs in the 3GPP library assumed signal carrier frequency was a known value and did not provide any design approach for control loop recovery of transferred data and carrier frequency. In ADS2005A, an RF receiver for recovering 3GPP CPICH symbols and carrier using a PLL are introduced to show transferred data recovery capability.

In the *RF_PLL_Examples_wrk*, the RF carrier is synchronized by a multi-rate PLL and CPICH data is recovered from the phase information on the RF carrier.

Basic Phase Lock Loop Operation

A typical PLL is shown below. A signal is sent to the reference input. The internal oscillator locks to the reference. Frequency and phase differences between the reference and internal sinusoid = k or 0. The internal sinusoid then represents a filtered version of the reference sinusoid.

Basic Phase Lock Loop Structure



Non-Linear Phase Lock Loop

There are three important parts for phase lock loop:

- Phase detector. This is a nonlinear device whose output contains the phase difference between the two oscillating input signals.
- Voltage controlled oscillator. This is a nonlinear device that produces an oscillation whose frequency is controlled by a lower frequency input voltage.
- Loop filter. While this can be omitted (resulting in what is known as a first- order PLL) it is always conceptually there because PLLs depend on some kind of lowpass filtering in order to function properly.

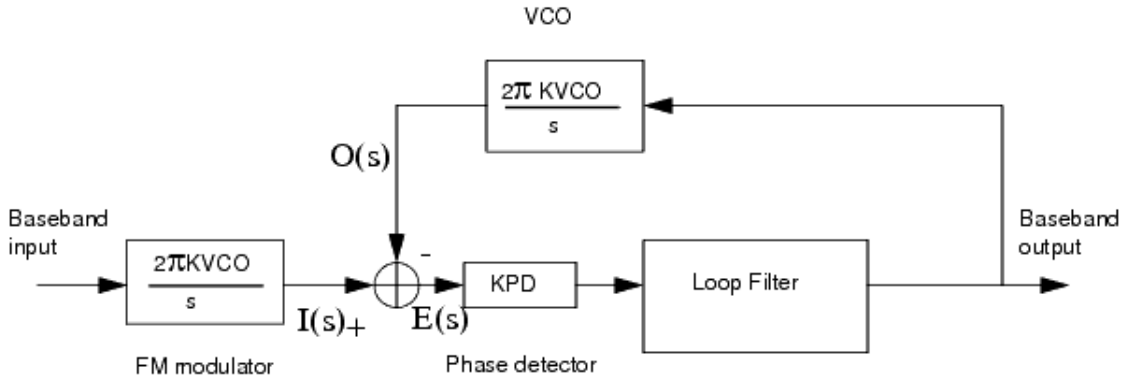
The phase detector takes as its input the reference signal and the output of the VCO. The output of the phase detector (the phase interference) is used as the control voltage for the VCO.

By appropriately selecting and setting these parts, the loop can be locked so that the frequency and phase interference between the input signal and the internal oscillator signal are close to zero.

Linear Approximation

Linear Approximation for RF PLL Control depicts a linear approximation to the nonlinear PLL shown in Basic Phase Lock Loop Structure. As you can see in this figure, the elements have been replaced with summers, amplifiers, a loop filter and an integrator; these are all linear elements. The network is a second-order control loop.

Linear Approximation for RF PLL Control



The loop filter transfer function can be expressed as

$$G(s) = KF\frac{(1+\tau_2 s)}{(1+\tau_1 s)}$$

where
KF = open loop gain
τ_1 = constant determined by RC parameters
τ_2 = constant determined by RC parameters

The closed loop transfer function for the linear approximation of the PLL can therefore be derived, giving the second-order loop transfer function:

$$\frac{O(s)}{I(s)} = \omega_n^2 \frac{\left(\frac{s}{\delta}+1\right)}{(s+2\zeta s\omega_n+\omega_n^2)}$$

where

$$\omega_n^2 = (KF)(KPD)\frac{(2\pi KVCO)}{\tau_1 s}$$

and

$$\xi = \frac{1}{2\omega_n \tau_1}(1 + KF(2\pi KVCO)KPD\tau_2)$$

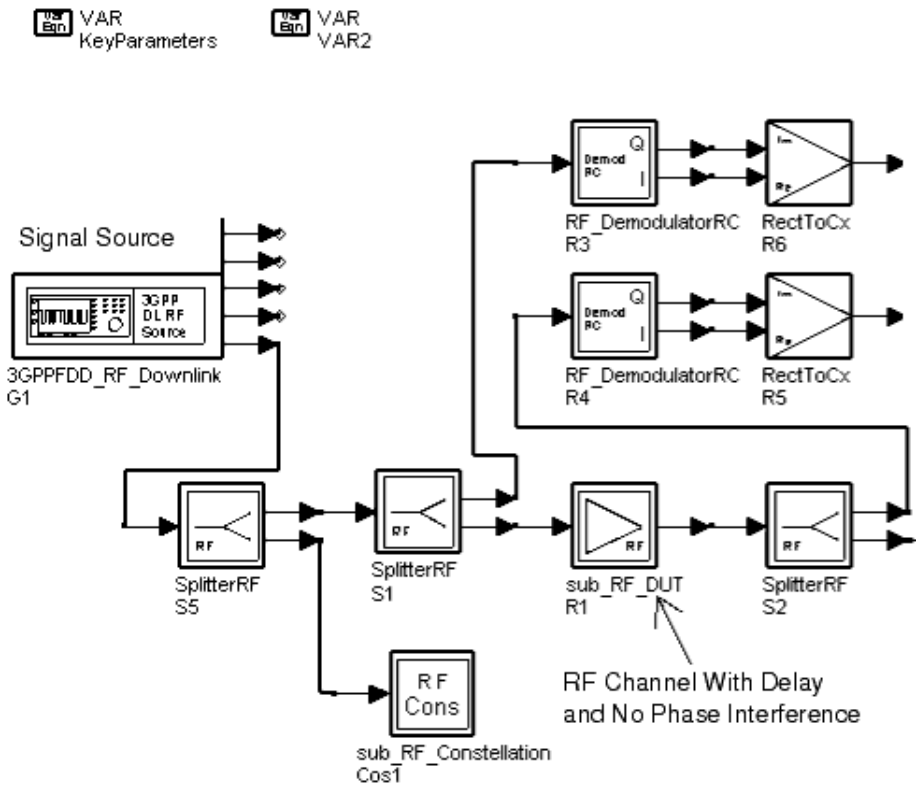
and
δ = 1/τ2.

Phase Lock Loop Applications in ADS

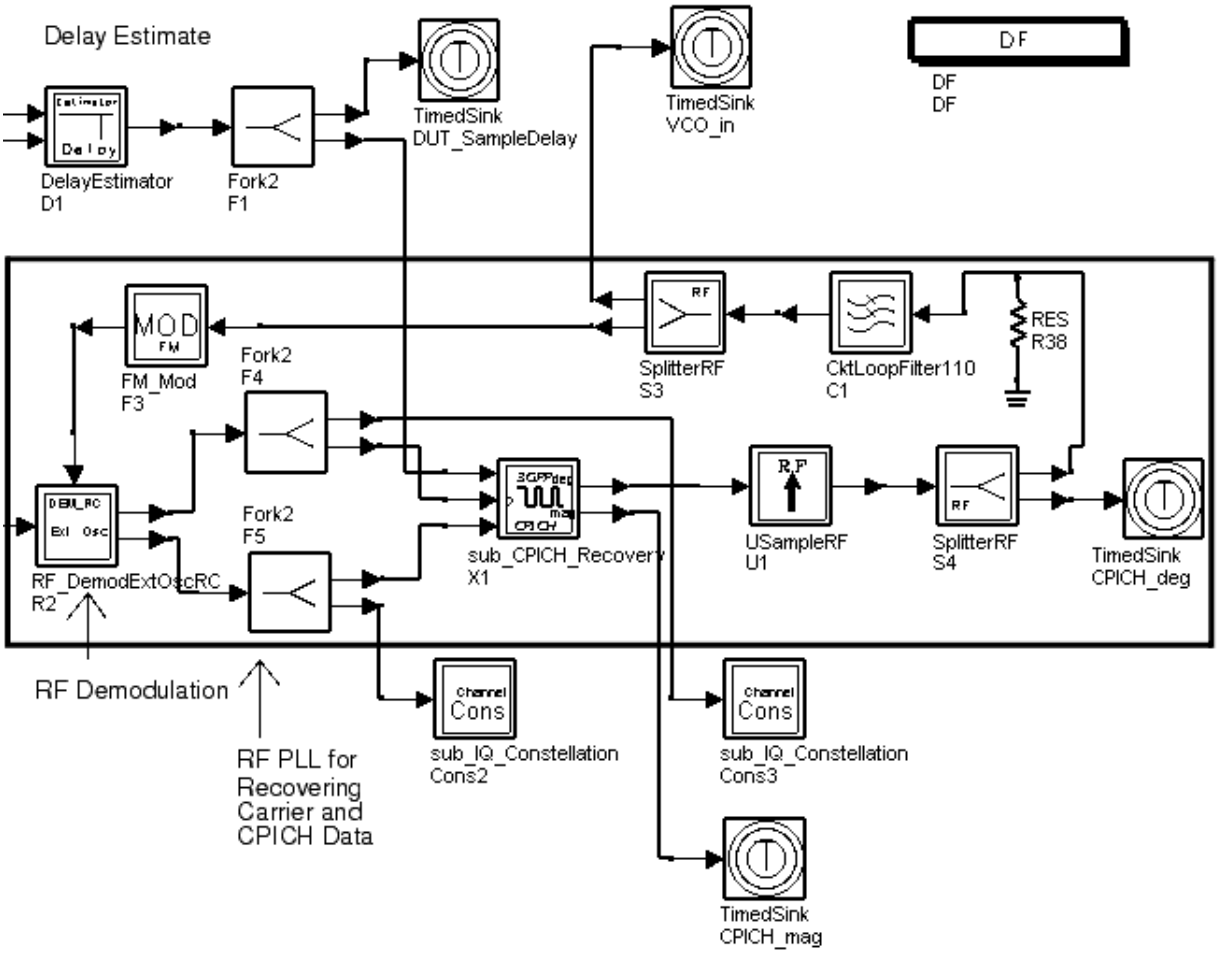
PLL can be used in various ways, including: carrier recovery, timing recovery, Costas loop, clock/data recovery, frequency synthesis, modulation/demodulation, and PLL application control problems (such as disk drive control, harmonic compensation, and motor control).

RF_Receiver_CPICH_PLL_check (RF_Receiver_CPICH_PLL_check Schematic (front-end) and RF_Receiver_CPICH_PLL_check Schematic (back-end)) provides a 3GPP receiver using PLL for recovering carrier frequency and demodulating the 3GPP data from the received signal. In the receiver, the PLL is used for recovering CPICH symbols and carrier frequency. In this design, the RF carrier is synchronized by a multi-rate PLL and CPICH data is recovered from the phase information on the RF carrier.

RF_Receiver_CPICH_PLL_check Schematic (front-end)



RF_Receiver_CPICH_PLL_check Schematic (back-end)



The 3GPPFDD_RF_Downlink source generates 3GPP downlink data. The 3GPP downlink signal is sent to an RF channel (sub_RF_DUT) with delay, phase interference, and nonlinear distortion. For an accurate sampling process, DelayEstimator is connected to the RF channel input and output through the RF_DemodulatorRC and RectToCx components for RF channel delay estimation.

To demodulate 3GPP data with carrier recovery capability, a PLL is designed using an RF demodulator with external oscillator and RC filter RF_DemodExtOscRC as a phase detector, sub_CPICH_Recovery for recovery of CPICH data including magnitude and phase, a circuit lowpass filter as a loop filter (CktLoopFilter110), and an FM modulator as VCO. Upsampler USampleRF is used in the PLL for data rate matching. In the loop, the CPICH phase difference is sent from sub_CPICH_Recovery to the loop filter; the loop filter can be used to extract the average phase difference. The loop will drive the phase difference toward zero and then lock the carrier frequency.

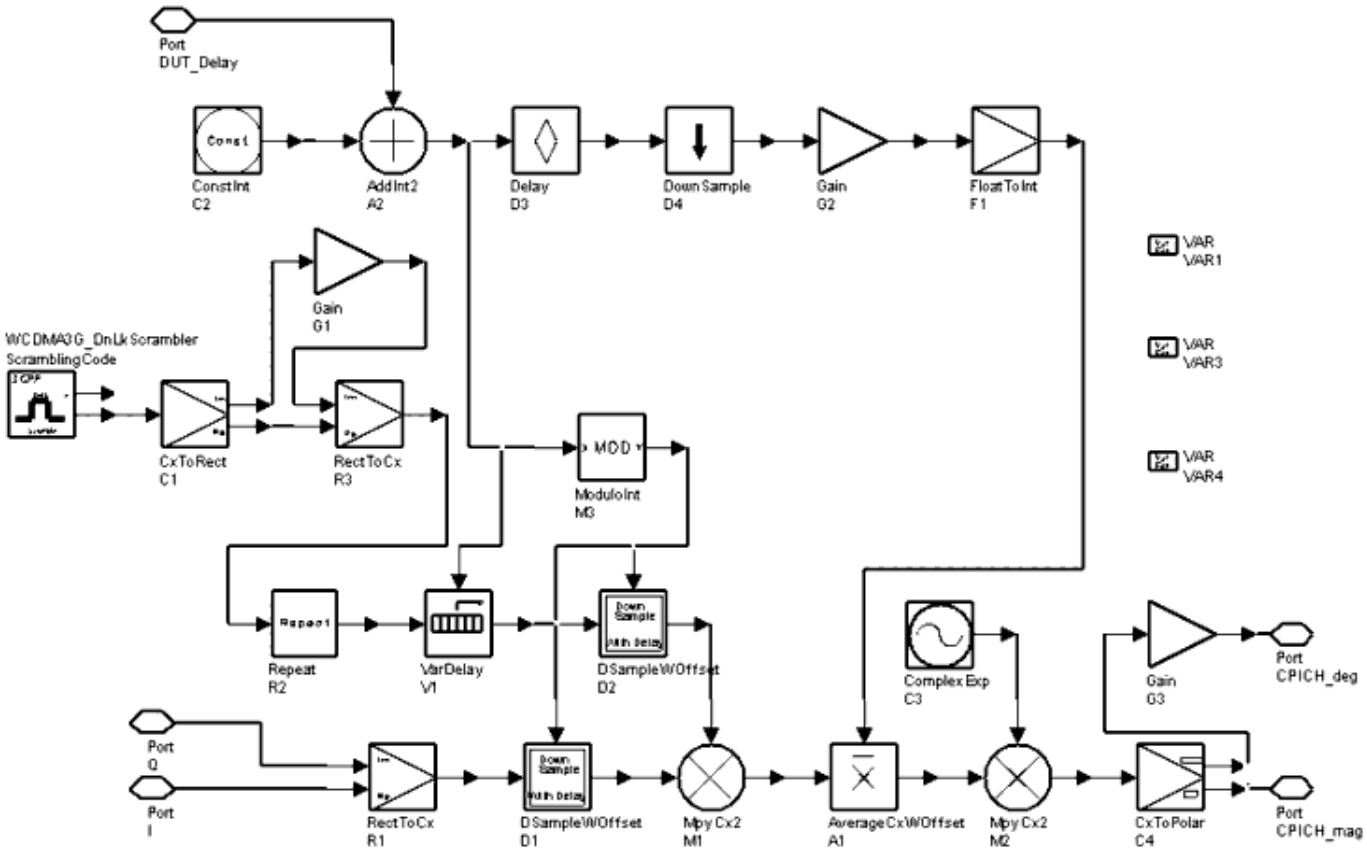
sub_CPICH_Recovery subnetwork (schematic is shown in sub_CPICH_Recovery Schematic) downsamples output data from RF demodulator RF_DemodExtOscRC (using channel delay information) and descrambles 3GPP CPICH data. I and Q data from RF_DemodExtOscRC are output in this subnetwork and converted to complex data through RectToCx. DSampleWOffset (D1) then downsamples the complex input data with RF channel delay information from Input 3 of sub_CPICH_Recovery through AddInt2 for combining additional delay from modulation and demodulation filters and ModuloInt for detecting the remaining portion of the module by SampPerChip.

WCDMA3G_DnLnkScrambler descrambles 3GPP CPICH data. DSampleWOffset (D2)

downsamples the CPICH scrambled data with RF channel delay information. MpyCx2 (M1) multiplexes the downsampled input and scramble code for descrambling CPICH data. AverageCxWOffset then performs a moving average to the descrambled CPICH data.

ComplexExp, MpyCx2 (M2), Gain (G3), and CxToPolar output magnitude and angle. The CPICH_deg output is used for the PLL to lock the carrier frequency; the CPICH_mag output is used to observe the CPICH magnitude.

sub_CPICH_Recovery Schematic

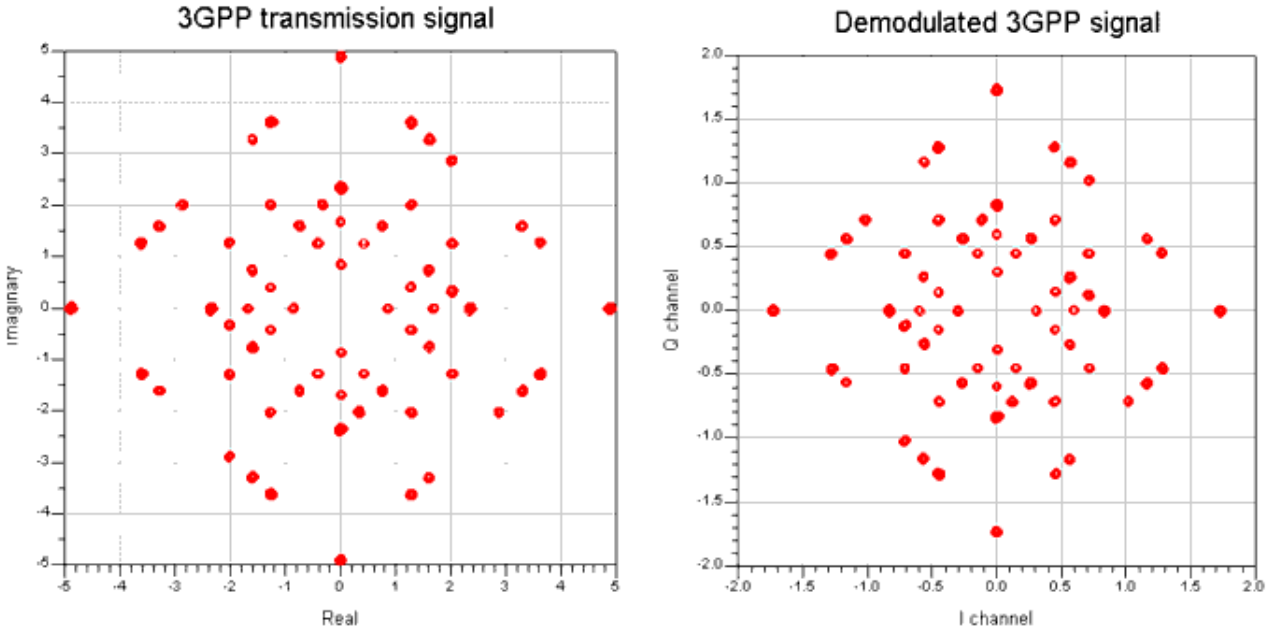


The parameter setup may follow the second-order loop procedure as discussed previously in Linear Approximation.

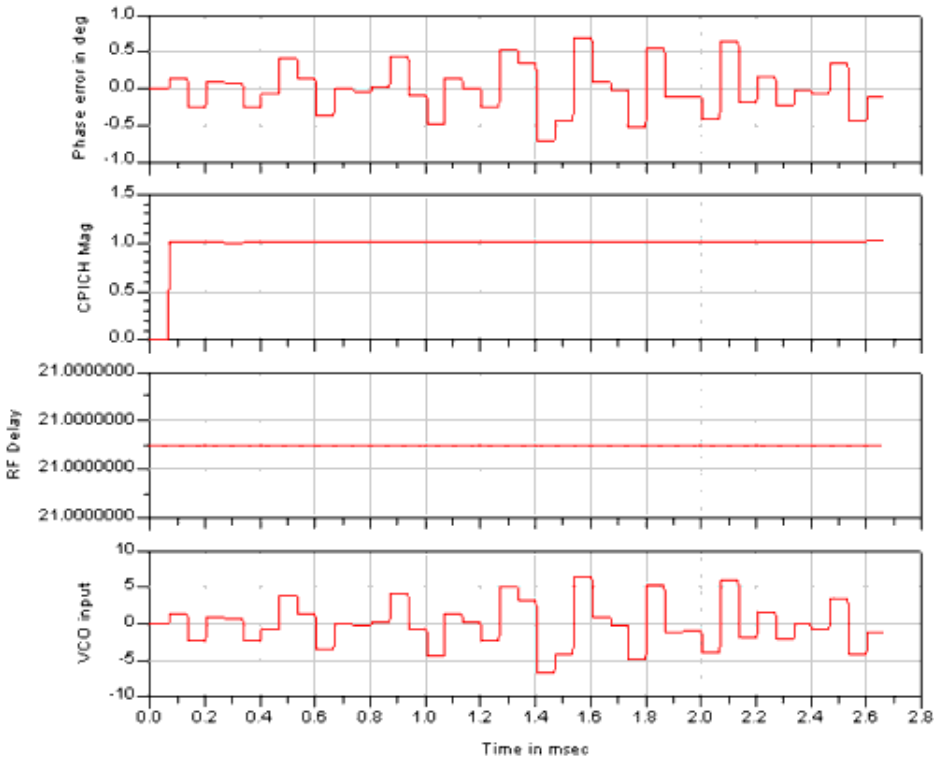
The sensitivity of the phase detector have been set to the value of 1 while the FM modulator sensitivity is set equal to the sensitivity of the VCO value of KVCO = 4.5 kHz/V. The capacitor value C is set to 20 pF.

For an RF channel with delay and no phase interference, test bench measurements are collected in *RF_PLL_Phase_Mag_noPhaseError.dds* for displaying PLL data and *RF_PLL_Constellation_Delay.dds* for showing 3GPP constellations. Demodulated 3GPP Data vs. Original Transmitted 3GPP Data shows that the PLL successfully locks the carrier and the demodulated signal constellation is then matched with the transmission constellation. PLL Data for an RF Channel Without Phase Interference shows the CPICH demodulated output magnitude closest to constant 1, just the same as the original CPICH data. And the PLL phase is locked in a small range from -1 to 1 degree.

Demodulated 3GPP Data vs. Original Transmitted 3GPP Data



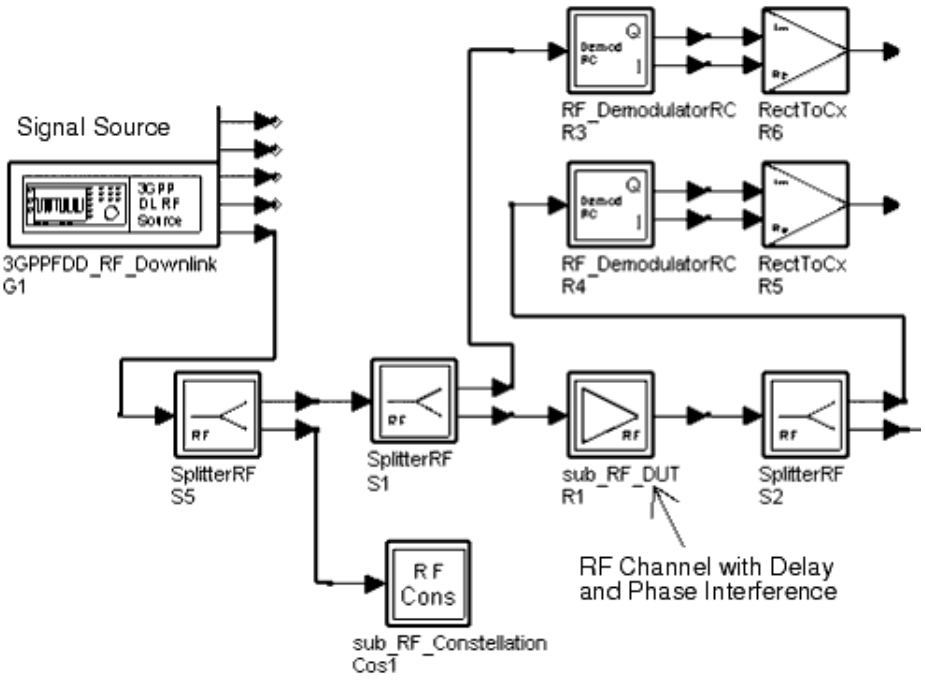
PLL Data for an RF Channel Without Phase Interference



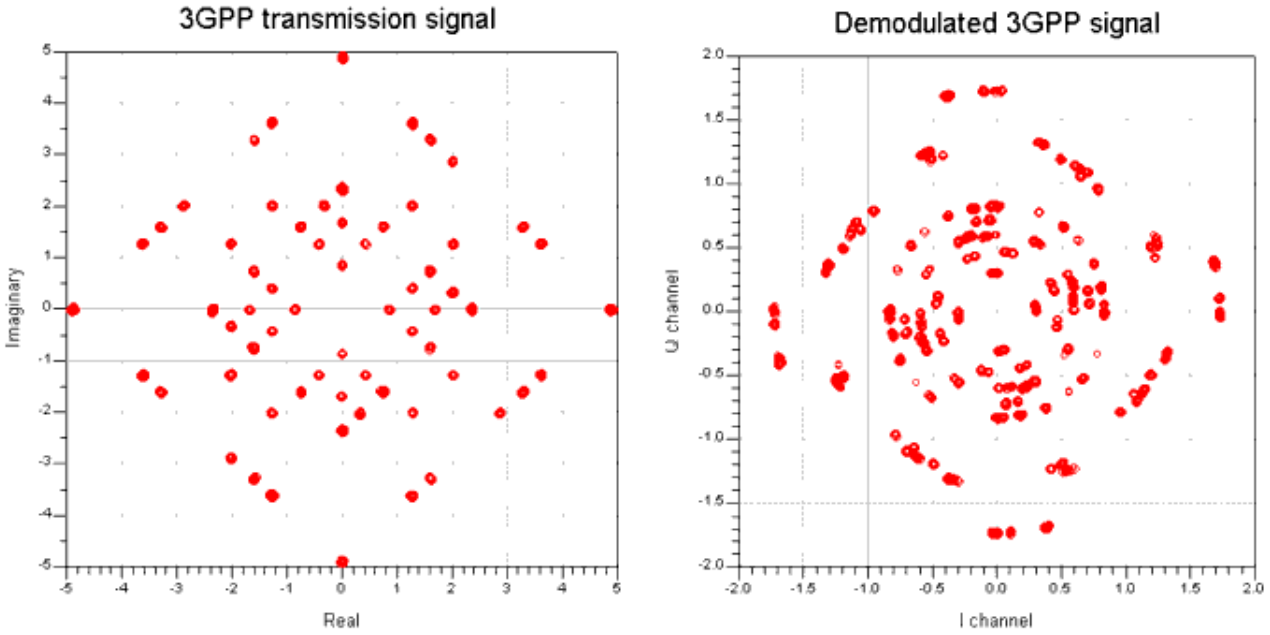
In *RF_Receiver* CPICP_PLL_test_, the RF channel with delay and phase interference is considered (sub_RF_DUT subnetwork in PLL Data for RF Channel with Delay and Phase Interference).

Measurements are collected in *RF_PLL_Phase_Mag_PhaseError.dds* for displaying PLL data and *RF_PLL_Constellation_PhaseError.dds* for showing 3GPP constellations. Demodulated vs. Original 3GPP Data for Channel Phase Interference shows that the PLL locks the carrier and the demodulated signal constellation is then reasonably matched with the transmission constellation. Internal PLL Data for an RF Channel with Phase Interference shows the CPICH demodulated output magnitude closest to constant 1, just the same as the original CPICH data. And the PLL phase is locked in a range of -10 to 10 degrees.

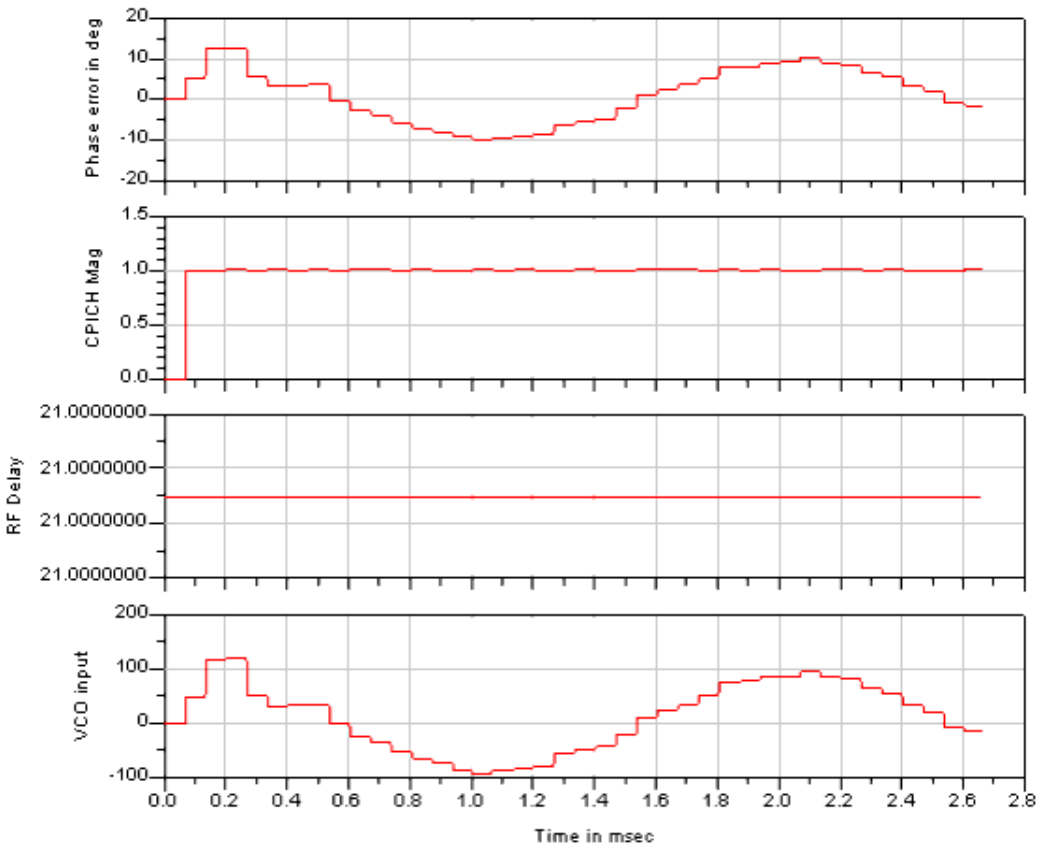
PLL Data for RF Channel with Delay and Phase Interference



Demodulated vs. Original 3GPP Data for Channel Phase Interference



Internal PLL Data for an RF Channel with Phase Interference



Automatic Gain Control Loop Designs

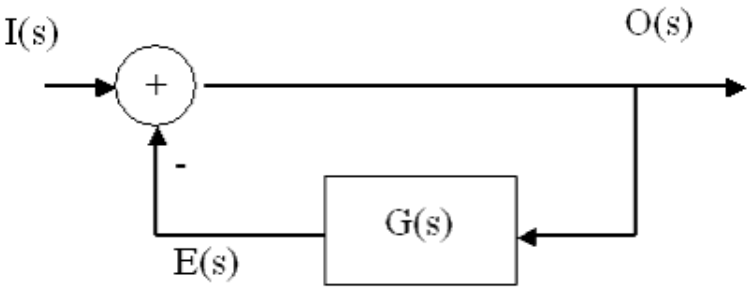
Automatic gain control loops are important in any communications system where wide amplitude variations in the output signal lead to performance degradation. These signals require good control to maintain a constant signal level at the output.

The structure of an AGC system is determined by requirements of the communication system. ADS provides a top-down RF design that can be used to analyze the steady-state and transient response of the control loop.

Linear AGC Control Loop

Though control loops can have general topologies, we can obtain useful design guidelines by studying the simple control loop shown below.

Simple Linear Control Loop



You can compare the performance of such a control loop with the performance of other more complex and nonlinear control loops. The Simple Linear Control Loop above shows an input signal I, and output signal O, an error signal, E, and an open loop transfer function, G(s).

The closed loop transfer function is derived as follows:

$$H(s) = \frac{O(s)}{I(s)} = \frac{G(s)}{1 + G(s)}$$

The characteristic of the open loop transfer function G(s) provides a classification that conveniently categorizes various control loop implementations. For our purposes, a second-order control loop is of interest. The loop *order* specifies the total number of poles in G(s). Additionally, the *type* of loop is specified by the number of poles at the origin in G(s).

The following discussion is for first- and second-order loops. The table below shows the basic control loop transfer functions. Parameter *K* in the open loop transfer function G(s) is called the loop gain.

Basic Control Loop Transfer Functions

Loop Order	Loop Type	Open Loop, G(s)	Closed Loop, H(s)
1	0	$\frac{K}{s+a}$	$\frac{K}{s+a+K}$
1	1	$\frac{K}{s}$	$\frac{K}{s+K}$
2	0	$\frac{Kbc(s+a)}{a(s+b)(s+c)}$	$\frac{Kbc(s+a)/a}{s^2+s(b+c+Kbc/a)+bc(1+K)}$
2	1	$\frac{Kb(s+a)}{a(s+b)s}$	$\frac{Kb(s+a)/a}{s^2+s(b+Kb/a)+Kb}$
2	2	$\frac{K(s+a)}{s^2}$	$\frac{K(s+a)}{s^2+sK+Ka}$

Control Loop Equations

First- and second-order linear control loops are often used as approximations for actual RF control loops. This section presents the equations defining these control loops in terms of their frequency- and time-domain performance parameters. The second-order, type 1 loop is discussed in detail; performance parameters of the other loops are presented in tables.

The second-order, type 1 closed loop response-in terms of loop natural frequency (ω_n), loop damping factor (ζ), and closed loop zero (δ)-can be expressed as:

$$H(s) = \frac{O(s)}{I(s)} = \frac{(s/\delta + 1)\omega_n^2}{s^2 + 2\varsigma\omega_n s + \omega_n^2}$$

where the following associations are made with the related expression from Basic Control Loop Transfer Functions.

$$\omega_n = \sqrt{Kb}$$

$$\varsigma = \frac{b + Kb/a}{2\sqrt{Kb}}$$

$$\delta = a$$

Loop Filters

The open loop transfer function G(s), can be represented in terms of a detector gain factor, KD, a voltage gain factor, KV, a loop filter, F(s), and possibly an integrator. The loop filter provides the poles and zeros of G(s).

Loop filters may be categorized on the basis of the filter order and number of filter zeros and integrators. The integrators are in fact poles and are included in the count of poles for the filter order. The loop filter gain is listed as K_F.

The loop filter is typically a circuit network composed of resistors, capacitors, and possibly, op amps. The op amp model typically includes a single dominant pole and time delay:

$$V_s = (V^+ - V^-)M\frac{e^{-j2\pi fT}}{1+j2\pi f/F}$$

where
V_s = op amp output voltage

V^+ = noninverting input voltage

V^- = inverting input voltage
M = magnitude of the voltage gain at dc
T = op amp time delay
F = frequency at which the gain magnitude is down by 3 dB
f = simulation frequency

Additionally, the op amp model includes input resistances, output resistance, and leakage resistances.

Though all of these op amp characteristics may be included within the control loop simulation, the following discussion on loop filters considers the op amp to have ideal infinite gain for simplification of the technical presentation.

The table below lists various loop filter designs, and associated parameters. The parameters *a*, *b*, and *c* are filter poles and zeros, and parameters τi are time constants.

Loop Filter Transfer Functions

Filter Design, Order, Zeros, and Integrators	Transfer Function with Poles and Zeros	Transfer Function with Time Constants
F100(s) 1,0,0	$\frac{K_F B}{s+b}$	$\frac{K_F}{s\tau_1+1}$
F101(s) 1,0,1	$\frac{K_F}{s}$	$\frac{K_F}{s}$
F110(s) 1,1,0	$\frac{K_F b(s+a)}{a(s+b)}$	$\frac{K_F(s\tau_1+1)}{s\tau_2+1}$
F111(s) 1,1,1	$\frac{K_F(s+a)}{as}$	$\frac{K_F(s\tau_1+1)}{s}$
F200(s) 2,0,0	$\frac{K_F bc}{(s+b)(s+c)}$	$\frac{K_F}{(s\tau_1+1)(s\tau_2+1)}$
F201(s) 2,0,1	$\frac{K_F b}{s(s+b)}$	$\frac{K_F}{s(s\tau_1+1)}$
F210(s) 2,1,0	$\frac{K_F bc(s+a)}{a(s+b)(s+c)}$	$\frac{K_F(s\tau_1+1)}{(s\tau_2+1)(s\tau_3+1)}$
F211(s) 2,1,1	$\frac{K_F b(s+a)}{as(s+b)}$	$\frac{K_F(s\tau_1+1)}{s(s\tau_2+1)}$

RF AGC Loop

Automatic gain control (AGC) is widely used in communication, radar, and instrumentation systems to maintain constant RF signal strength. AGC loops may be classified as coherent or incoherent.

- The coherent loops use an RF synchronous detector with a local oscillator (LO) that is synchronous with the input RF signal. The detector output voltage is typically proportional to the detector input RF voltage in phase with the LO signal. The LO synchronization is achieved by passing the input RF signal through a carrier recovery circuit.
- The noncoherent loops use an RF power detector with output voltage typically proportional with input RF power.

Though any AGC loop is inherently a nonlinear control loop, simplifying assumptions allows you to use a linear model for analysis of expected performance of the nonlinear loop.

System Level Modeling

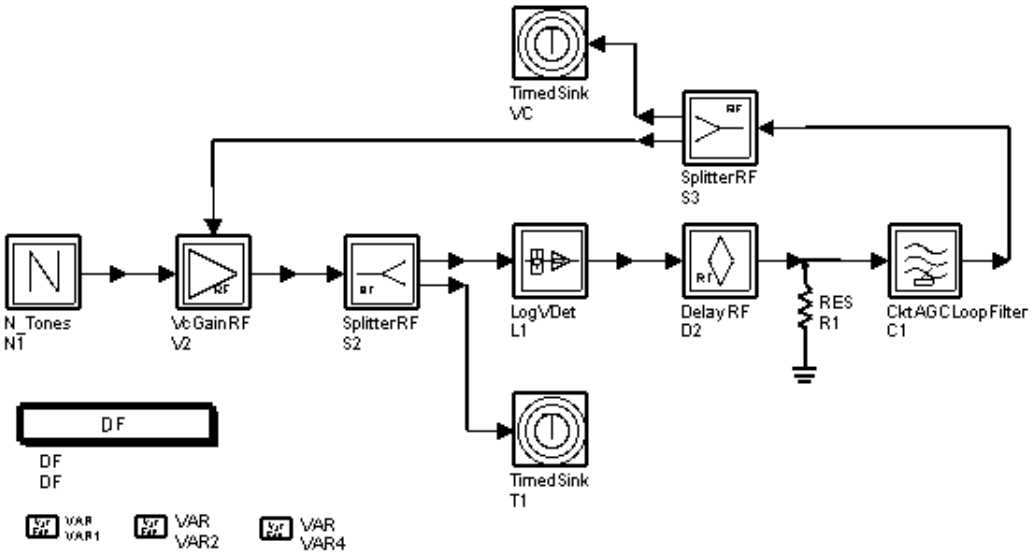
When modeling an RF system, we often include nonlinear and non-ideal factors of interest. However, the initial analysis of such systems is often based on a linear approximation. The next section discusses a nonlinear model and its linear approximation.

Nonlinear Model

A typical RF AGC loop includes a voltage- (or current-) controlled amplifier (VcGainRF), an amplitude detector (LogVDet), and a loop filter (CktAGCLoopFilter).

The figure below shows a typical RF AGC control loop.

Typical RF AGC Control Loop



The RF input signal (RFin) is amplified by the voltage-controlled RF amplifier (VcGainRF) that has a gain controlled by the output of the loop filter (CktAGCLoopFilter), and has a gain in dB that is typically proportional to the control voltage. The VcGainRF output through LogVDet is filtered by a bandpass filter (CktAGCLoopFilter) to limit the noise into the loop detector.

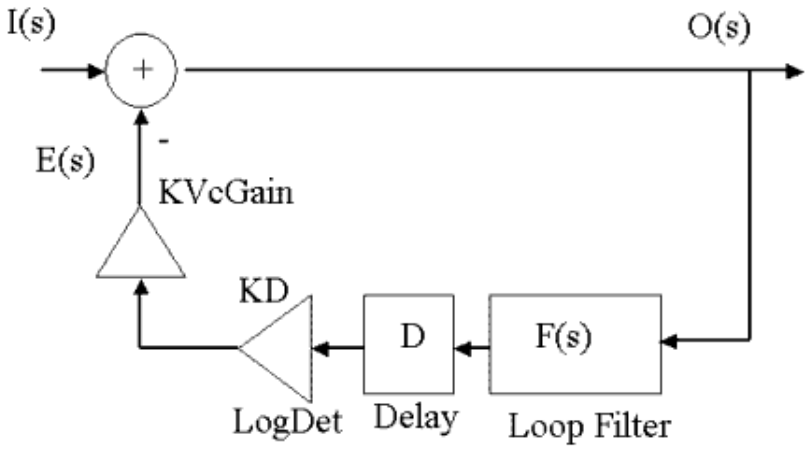
The detector output voltage is typically proportional to input RF power. The detector (LogVDet) has a sensitivity expressed in V/dB. The detector output is in the differential gain and then filtered by the loop filter, to obtain the control voltage VC, which closes the AGC loop.

The bandpass bandwidth is larger than the exact loop bandwidth so as to not interfere with loop operation. VcGainRF has gain *agc_gain* that is related to the control voltage VC, which is limited to the lower limit VMin and higher limit VMax for the VcGainRF model parameters.

Linear Approximation

The linear control loop shown in RF AGC Loop, Linear Model is a linear approximation to the nonlinear AGC loop shown in Typical RF AGC Control Loop. As you can see from this figure, the elements have been replaced with summers, amplifiers, a loop filter, and an integrator. These are all linear elements. This network is a second-order, type 1 control loop.

RF AGC Loop, Linear Model



The elements shown in RF AGC Loop, Linear Model are labeled in groups for the VcGain, LogDet, Delay, and Loop Filter. The VcGain linear model with KVcGain gain sensitivity in unit of V/dB. The LogDet model has a sensitivity, KD, given in units of V/dB. The detector provides a specific output voltage, B(V), for an input power level PN (dBm). The Loop Filter has the transfer function F(s).

Comparing RF AGC Loop, Linear Model and Typical RF AGC Control Loop, the following associations may be made: RFin(s) is I(s), RFout(s) is O(s).

When analyzing this linear control loop for its dynamic closed loop performance, the constant terms in RF AGC Loop, Linear Model may be ignored and the relationships between I(s), O(s), and E(s) can be derived as follows

E(s) = I(s) - O(s)

O(s) = KVcGain KD F(s)/s E(s)

F(s) = KF b (s+a) / a / (s+b)

from which the open loop, G(s), and closed loop, H(s), transfer functions for a second-order, type 1 loop are obtained:

$$G(s) = \frac{Kb(s+a)}{as(s+b)}$$

$$H(s) = \frac{Kb/a(s+a)}{s^2 + s(b + Kb/a) + Kb}$$

The AGC loop design may follow the second-order, type 1 linear control loop design procedure discussed in the section Second-Order, Type 1, Linear Control Loop Design.

K = −KVcGain KD G KF = KB = 4.3955 (10^4)

where

KVcGain KD G = 1000

KVcGain = 0.025 V/dB

KD = 0.025 V/dB

G = 1000/KVCA/KD = 20,000

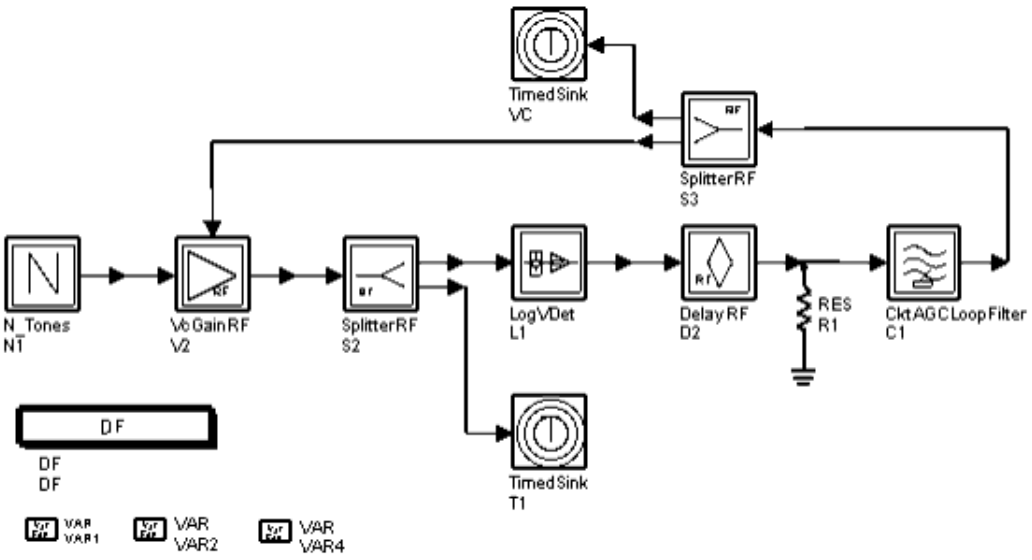
KF = −KB/1000

and KB, a, b, R1, C1, R2, R3 are as described in the section Second-Order, Type 1, Linear Control Loop Design.

RF AGC Loop Performance

In example workspace *RF_AGC_Loop_wrk, RF_AGC_loop_test* shown in RF_AGC_loop_test Schematic provides an AGC loop for an amplifier with a communication input signal.

RF_AGC_loop_test Schematic



In RF_AGC_loop_test Schematic, the control voltage feedback is sent to the voltage

controlled gain block VcGainRF with gain dB/V. A delay block DelayRF in the loop is required to resolve dataflow deadlock. A 50-ohm resistor at the DelayRF output is required because the CktAGCLoopFilter has infinite input resistance.

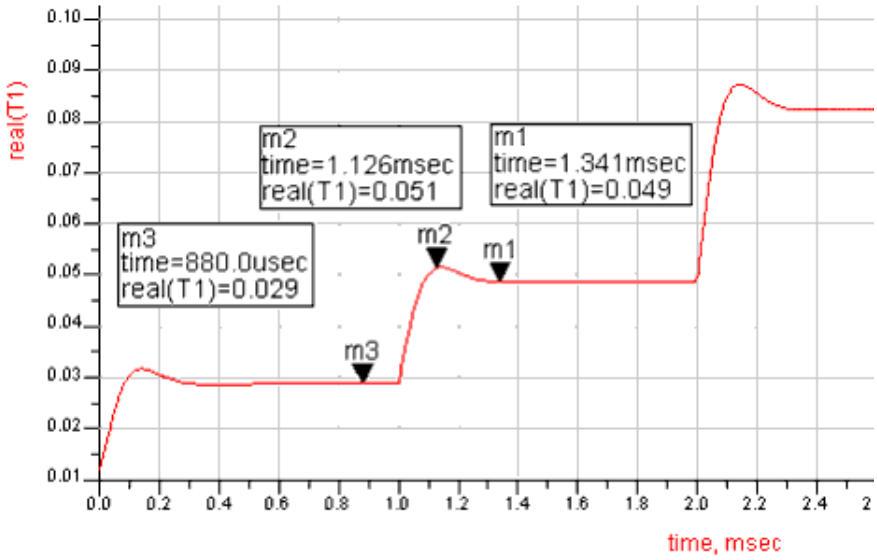
- Power into the loop from the N_Tones source is fixed at -30 dBm.
- LogVDet sensitivity is in terms of 0.025 volts/dB.
- Gain outside CktAGCLoopFilter is 0.025.

The pwl() expression sets the desired OpAmp reference voltage RefV as a function of time. This defines the power level desired for the AGC output that can be measured by TimedSink T1. The AGC output power level is independent of input power level sent by the signal source N_Tones. This means the design has the desired AGC characteristics.

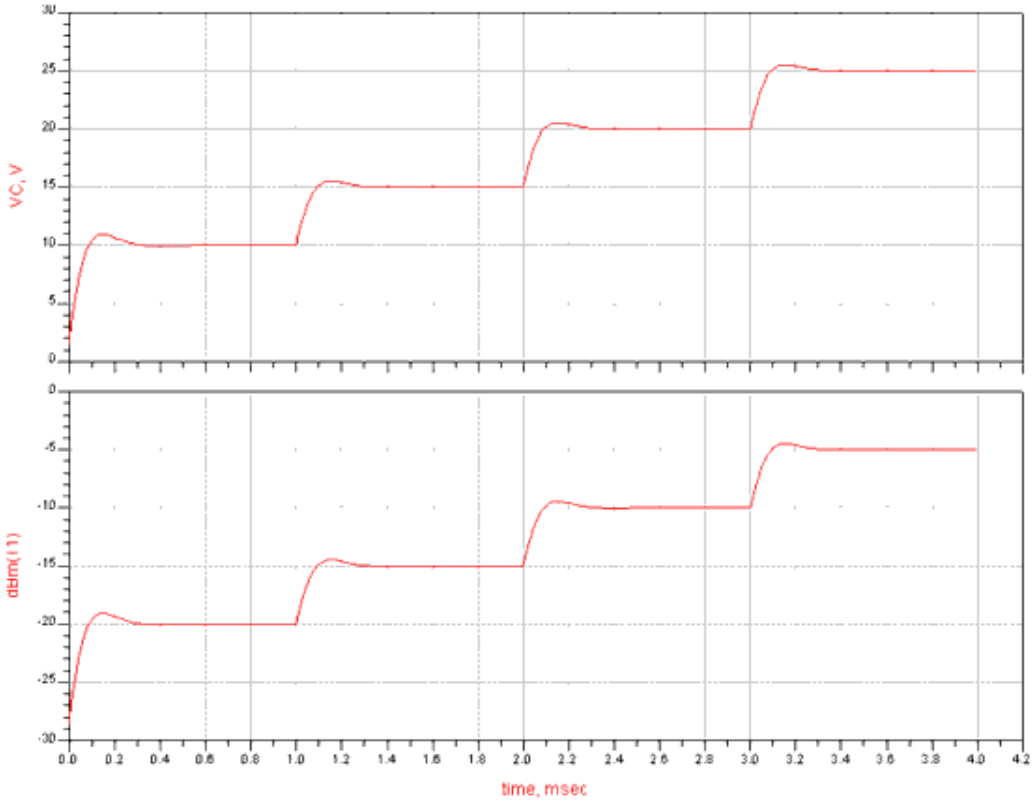
To check the parameter settings for the AGC loop, in Step Response for the AGC Loop the step response is displayed. From this figure, overshooting is 10%; this is what we expect as predicted in Second-Order, Type 1, Linear Control Loop Design.

AGC Loop Simulation Results shows the simulation results. The upper curve is the resultant control voltage applied to the VcGainRF block; the lower curve is the power level into TimedSink T1.

Step Response for the AGC Loop



AGC Loop Simulation Results



Second-Order, Type 1, Linear Control Loop Design

The design of a second-order, type 1 control loop may begin with various specifications in the frequency domain or in the time domain. For this discussion, assume the following specified parameters:

ζ loop damping factor, 0.707

δ* loop normalized zero, 1.0

BW closed loop −3 dB frequency, 4.5 kHz

From these known values, we obtain the desired open loop and closed loop transfer function parameters from the equations presented in the section Control Loop Equations. Let the open loop transfer function be GB(s). The open and closed loop parameters are as follows:

$\omega n = 2\pi BW/\sqrt{A+\sqrt{A^2+1}}$; $A=1+1/\delta^{*2}-2\zeta^2$

$= 1.8197\ (10^4)$

$a = \delta^* \omega n$

$= 1.8197\ (10^4)$

$b = 2\zeta w_n - \omega n^2/\delta = \omega n\ (2\zeta - 1/\delta^*)$

= 7.5335 (10^3)

KB = $\omega n^2 / b$

= 4.3955 (10^4)

τ1 = zero time constant 1/a

τ2 = pole time constant 1/b

If GB(s) is composed of a gain of 1000 in cascade with an integrator and a loop filter of the form F110(s), as shown in *Loop Filter Transfer Functions*, then K_B, τ1 and τ2 are used to obtain the values for R1, R2, R3, and C1 as follows

$$\tau_2 = \tau_1 + \frac{K_B}{1000} C_1 R_1$$

$$R_1 = \frac{(\tau_2 - \tau_1)1000}{K_B C_1}$$

$$R_2 = \tau_1 / C_1$$

$$R_3 = \frac{K_B}{1000} R_1$$

Let

C1 = 200 pf

Then,

R1 = 8848 Ω

R2 = 274.8 KΩ

R3 = 388.9 KΩ

From the values of ωn, ζ, and δ, we obtain the following theoretical frequency- and time-domain response parameters:

FP = 1.86 kHz

AP = 0.817 dB

BW = 4.50 kHz

BWn = 6.43 kHz

t*ms = 2.718 sec

tms = 0.149 msec

POs = 10.74%

The open loop function GB(s) has the form F211(s). Let GA(s) represent the approximately equivalent open loop transfer function that has the form F210(s), as discussed in the previous section. Let GA(s) have a pole that is 100 times smaller than b in place of the integrator in GB(s).

$$GB(s) = \frac{K_B b(s+a)}{as(s+b)}$$

$$GA(s) = \frac{K_A bc(s+a)}{a(s+b)(s+c)}$$

where
$K_A = K_B / C$
$c = b/100$

The step input response for the closed loops with open loop gain GA(s) and GB(s) is shown in Closed Loop Response for Step Input Alternate Second-Order Loop Designs GA(t) for Second-Order, Type 0 GB(t) for Second-Order, Type 1. You can see in this figure that the time-domain transient response is nearly identical.

Closed Loop Response for Step Input Alternate Second-Order Loop Designs GA(t) for Second-Order, Type 0 GB(t) for Second-Order, Type 1

References

1. M. Jeruchim, P. Balaban, K. Shanmugan "Simulation of Communication System" Plenum Press, 1992.

Timed Data Processing Components

The Timed Data Processing library contains time domain components for processing bits and symbols including IQ coders and symbol format converters.

Each timed data processing component consumes and produces baseband timed signals. If a component receives another class of scalar signal, the received signal is automatically converted to a timed (baseband) type of signal. Auto conversion from complex scalar signal class to timed class is not allowed; the user must explicitly use a signal converter. These components do not accept any matrix class of signal.

These components only accept a baseband timed signal and any received RF (complex envelope) timed signals is first transformed into its baseband equivalent form before use by the component.

An RF (complex envelope) timed signal is converted to its equivalent baseband form as follows:

$$V_{bb}(t) = Re\left\{(v_{RF}(t))e^{j2\pi f_c t}\right\} = Re\left\{(v_I(t) + jv_Q(t))e^{j2\pi f_c t}\right\}$$

where
$V_{bb}(t)$ is the total representation for the RF signal (also called the baseband representation)
$v_{RF}(t)$ is the RF signal complex envelope at characterization frequency fc (also called the equivalent complex baseband envelope representation for the RF signal)
$v_I(t)$ is the RF timed signal in-phase envelope

$v_Q(t)$ is the RF timed signal quadrature-phase envelope

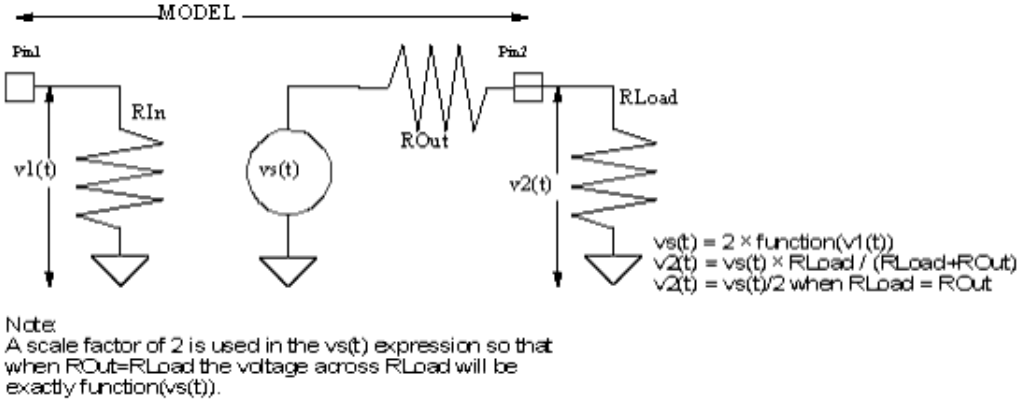
f_c is the RF signal characterization frequency

For this equivalence to be valid, the simulation time step must be less than the inverse of the characterization frequency and the RF signal information content has an information bandwidth less than the RF characterization frequency.

All data processing components have user-specified input resistance (RIn) and output resistance (ROut). Input resistance is for a shunt resistor at each component input pin; output resistance is for a series resistor at each component output pin.

Each component internal output signal, vs(t), has a value equal to twice the output pin signal, v2(t), when the output is connected to a matched resistor load. The circuit model illustrated in 2-Port Circuit Model for a 2-port (1 input, 1 output) timed linear component demonstrates this relationship.

2-Port Circuit Model



The output pin signal v2(t) at the output series resistance is dependent on the value of the load resistance connected to it. When the load resistor RLoad is equal to the model output resistor ROut the value of v2(t) is equal to vs(t)/2; otherwise, based on the voltage divider action, v2(t) is:

$v2(t) = vs(t) \times RLoad/(RLoad + ROut)$

The input and output resistor values must be greater than 0 ohm.

The input and output resistors contribute additive thermal noise power (kTB) to the output signal when the specified resistance temperature (RTemp) is greater than absolute zero (−273.15°C) where:

k = Boltzmann's constant
T = temperature in Kelvin
B = simulation frequency bandwidth:
1/2/tstep if signal is a timed baseband signal;
1/tstep if signal is a timed complex envelope signal

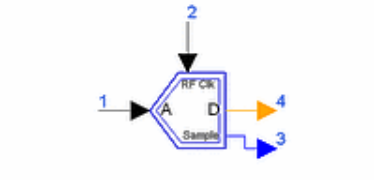
When RTemp > −273.15, the noise contributed from each resistor instance is an independent noise process. This noise is dependent on the DefaultSeed value in the DF (data flow) controller. When DefaultSeed=0, then the noise generated for each simulation is different. When DefaultSeed>0, then the noise generated for each simulation, though random, has the same initial seed starting condition and thus results in reproducible noise for each simulation.

Note
Information regarding time domain signal differences between ADS Ptolemy simulations and Circuit Envelope and Transient simulations is given in the *Timed Synchronous Dataflow* (ptolemy) section in the *ADS Ptolemy Simulation* (ptolemy) documentation.

Components

- *ADC Timed* (timed)
- *AtoD* (timed)
- *AtoD ADI* (timed)
- *BinaryCoder* (timed)
- *BinaryCombiner* (timed)
- *BinaryCombinerClocked* (timed)
- *DAC Timed* (timed)
- *DQPSK Pi4Decoder* (timed)
- *EncoderIQ* (timed)
- *EncoderIQClocked* (timed)
- *Interpolator* (timed)
- *QPSK Decoder* (timed)
- *SymbolConverter* (timed)
- *SymbolConverterClocked* (timed)
- *SymbolSplitter* (timed)
- *SymbolSplitterClocked* (timed)

ADC_Timed



Description: Analog-to-digital converter with RF clock for analog input sample-and-hold
Library: Timed, Data Processing
Class: TSDF_ADC_Timed
Derived From: _SampleAndHoldRF

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
NBits	number of bits	8		int	[2:16]
VRef	reference voltage, -VRef<=input<=VRef	1.0	V	real	(0, ∞)
INL	integral nonlinearity relative to least significant bit (LSB)	0.0		real	[DNL/2, ∞)
DNL	differential nonlinearity relative to least significant bit (LSB)	0.0		real	[0, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed
2	clock	sampling clock	timed

Pin Outputs

Pin	Name	Description	Signal Type
3	Asample	output sampled analog baseband signal	real
4	D	output binary data NBit word in integer form	int

Notes/Equations

1. This component models an analog-to-digital converter with integral and differential nonlinearities. The input is an analog baseband/RF signal. The output is digital word in integer form.

   The input signal is sampled at the rising zero crossing of the RF clock signal. A rising zero crossing is detected when a negative voltage clock sample is followed by a positive voltage clock sample. Once a rising zero crossing is detected, linear interpolation is used to find the exact zero crossing time as well as the value of the input signal at the zero crossing time.

   The N_Tones source is typically used as the RF_Clock signal. In order to make the zero crossing detection more stable, make sure that the clock signal is not sampled at points where its value is close to 0. For example, if there are 4 samples per clock period, and the Phase of N_Tones is set to 0.0, then the four samples in the clock period are going to be 1.0, 0.0, -1.0, 0.0. However, due to the finite precision used in

computer arithmetic 0.0 might actually not be exactly 0.0 but a very small value just greater or lower than 0.0 (e.g. 2.9478e-15 or -8.177e-15). To avoid this problem you can set the Phase parameter of N_Tones to 45 (or any other odd multiple of 45). In this case, the four samples in the clock period are going to be 0.707, -0.707, -0.707, 0.707.

Note
This component does not downsample the input signal; instead, it samples and holds the input signal. To downsample the output of this component, use an external DownSample component.

2. To avoid clipping, the input signal must be in the range of (-VRef, VRef).
3. DNL (differential nonlinearity) error is defined as the difference between an actual step width and the ideal value of 1 LSB (least significant bit, 1 LSB = 2 × VRef / 2^{NBits}). For an ideal analog-to-digital converter, in which the DNL=0 LSB, each analog step equals 1 LSB. The DNL parameter is used to set the maximum value of DNLs. A DNL error specification of less than 1 LSB guarantees a no missing codes transfer function.

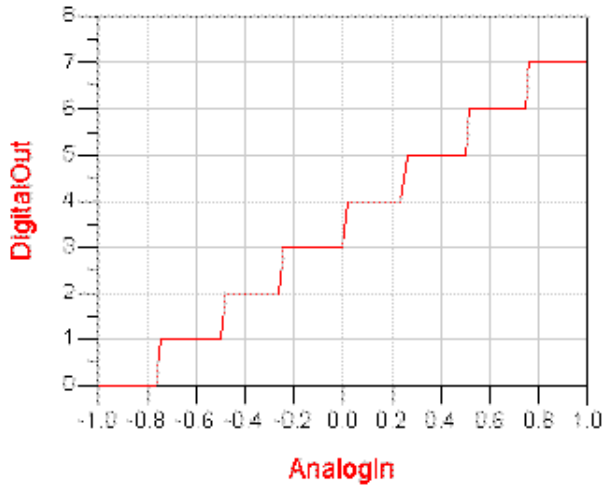
Note
There is no guarantee that the value of the DNL parameter will be reached. The DNL error is modeled by a normal (Gaussian) distribution. The distribution has an approximate 1% probability that the DNL error will be equal to or greater than the DNL parameter value (or less than -DNL). Those numbers are then truncated to DNL (or -DNL) before further processing.

4. INL (integral nonlinearity) error is defined as the deviation, in LSB, of the analog-to-digital converter actual transfer function from a ideal straight line. The INL parameter is used to set the maximum value of INLs.

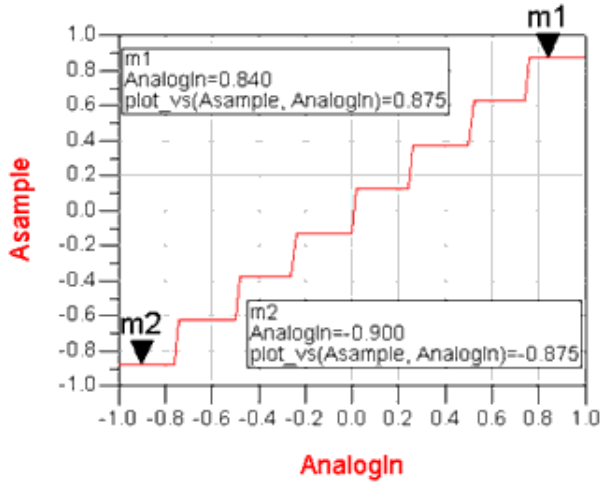
Note
There is no guarantee that the value of the INL parameter will be reached. It depends on the NBits, INL, and DNL parameter values. With larger NBits, smaller INL, and larger DNL, the INL will be reached more easily and frequently.

5. For an ideal ADC (INL=0 and DNL=0) the conversion threshold is {-VRef + i × LSB}, where i = 1, ... , $2^{NBits} - 1$.
6. For an ideal ADC (INL=0 and DNL=0) the Asample output is in the range of [-(VRef - 0.5 LSB), (VRef - 0.5 LSB)] with value {-VRef + (i - 0.5) × LSB}, where i = 1, ... , 2^{NBits} .
7. DigitalOut vs. AnalogIn and Asample vs. AnalogIn show the output of an ideal ADC (INL=0 and DNL=0) with NBits=3, VRef=1.
   In this case, LSB = 0.25; conversion threshold is {-0.75, -0.5, -0.25, 0.0, 0.25, 0.5, 0.75}; and Asample is {-0.875, -0.625, -0.375, -0.125, 0.125, 0.375, 0.625, 0.875}.

DigitalOut vs. AnalogIn

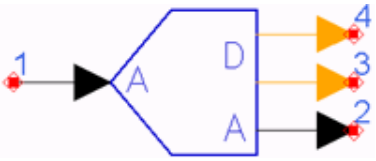


Asample vs. AnalogIn



8. For information regarding timed data processing component signals, refer to the *Timed Data Processing Components* (timed).

AtoD



Description: Analog to Digital Converter with Integral and Differential Nonlinearities
Library: Timed, Data Processing

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0,∞)
ROut	output resistance	DefaultROut	Ohm	real	(0,∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15,∞)
NBits	Number of bits	8		int	[2,16]
VRef	Reference voltage, -VRef<=input<=VRef	1.0	V	real	(0,∞)
INL	Integral nonlinearity relative to least significant bit (LSB)	0.0		real	[DNL/2,∞)
DNL	Differential nonlinearity relative to least significant bit (LSB)	0.0		real	[0,∞)
ConversionType	Type of input conversion: Clocked, Downsampled	Downsampled		enum	
Clock	Internal cosine clock frequency (used when ConversionType=Clocked)	0.2e6	Hz	real	(0,∞)
Phase	Internal clock phase (used when ConversionType=Clocked)	0.0	deg	real	(-∞,∞)
DownsampleFactor	Downsampling ratio (used when ConversionType=Downsampled)	1		int	[1,∞)
DownsamplePhase	Downsampling phase (used when ConversionType=Downsampled)	0		int	[0,Factor-1]
AntiAliasingFilter	Turn off/on anti-aliasing filter before downsampling (used when ConversionType=Downsampled): OFF, ON	OFF		enum	
ExcessBW	Excess bandwidth of raised cosine anti-aliasing filter (used when ConversionType=Downsampled and AntiAliasingFilter=On)	0.5		real	[0,1]
EnableJitter	Enable jitter: NO, YES	NO		enum	
RJrms	Standard deviation of random jitter	0.0	sec	real	[0,∞)
OutputDigitalFormat	Output digital format: Offset binary, Twos-complement	Twos-complement		enum	

Pin Inputs

Pin	Name	Description	Signal Type
1	A_in	input analog signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
2	A_out	output sampled analog baseband signal	timed
3	D_I	output NBit word as integer for I channel	int
4	D_Q	output NBit word as integer for Q channel	int

Notes/Equations

1. This component models an analog-to-digital converter with integral and differential nonlinearities. The input at A_in can be a real baseband signal or a complex envelope signal. The output digital words are in integer form. When the input is a real baseband signal, the output digital word is at the D_I output. When the input is a complex envelope signal, the D_I output contains the digital word for the I envelope and the D_Q ouput contains the digital word for the Q envelope. The A_out contains the quantized form of the input signal. If A_in is a complex envelope signal, then the A_out is also a complex envelope signal at the same characterization frequency.
2. When the ConversionType = Clocked, or the ConversionType = Downsampled and the DownsampleFactor = 1, then this block reads 1 sample from the input and writes 1 sample to the outputs.
3. When the ConversionType = Downsampled and the DownsampleFactor = N, then this block reads N samples from the input and writes 1 sample to the outputs.
4. Summary operation when ConversionType = Clocked
   The *Clock* and *Phase* values define an internal clock whose positive zero crossing results in sampling the input A_in and quantization of that input resulting in the output values. The clock zero crossings are inclusive of any specified random jitter (*RJrms*) applied to the clock. Between zero crossings of the internal clock, the outputs are held constant.
5. Summary operation when ConversionType = Downsampled
   The *DownsampleFactor*, *DownsamplePhase*, *AntiAliasingFilter*, and *ExcessBW* values define the downsampler operation on the input A_in with quantization of that downsampled input resulting in the output values. The input A_in has any specified random jitter (*RJrms*) applied to it before downsampling.
6. Summary operation of the quantization of the input A_in
   The *NBits*, *VRef*, *INL*, *DNL*, and *OutputDigitalFormat* values are used to quantize the input A_in_ to generate the digital words that appear at the integer outputs D_I and D_Q and the quantized analog output A_out. The input A_in is limited to values within the range of [- *VRef*, *VRef*] and clipped to -*VRef* or *VRef* when outside that range. When *OutputDigitalFormat* is *OffsetBinary*, the digital words span integer values in the range [0, 2^ *NBits* - 1]. When *OutputDigitalFormat* is *Twos-complement*, the digital words span integer values in the range [-2^(*NBits* - 1), 2^(*NBits* - 1) - 1]. The *INL* and *DNL* value affect the quantization levels as described in the following notes.
7. DNL (differential nonlinearity) error is defined as the difference between an actual step width and the ideal value of 1 LSB (least significant bit, 1 LSB = 2 × VRef / 2^{NBits}). For an ideal analog-to-digital converter, in which the DNL=0 LSB, each analog step equals 1 LSB. The DNL parameter is used to set the maximum value of DNLs. A DNL error specification of less than 1 LSB guarantees a no missing codes transfer function.

   Note
   There is no guarantee that the value of the DNL parameter will be reached. The DNL error is modeled by a normal (Gaussian) distribution. The distribution has an approximate 1% probability that the DNL error will be equal to or greater than the DNL parameter value (or less than -DNL). Those numbers are then truncated to DNL (or -DNL) before further processing.

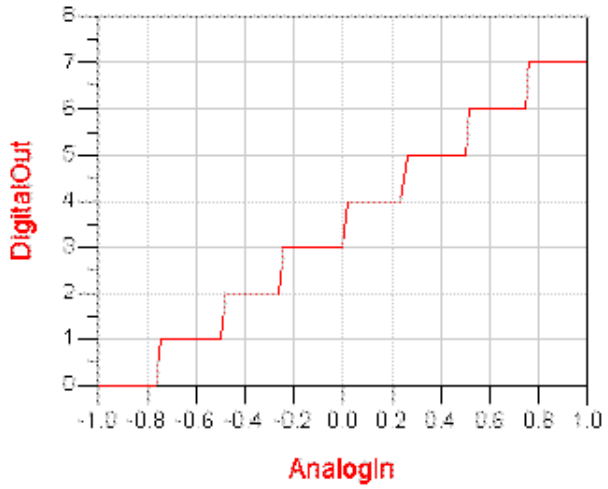
8. INL (integral nonlinearity) error is defined as the deviation, in LSB, of the analog-to-

digital converter actual transfer function from a ideal straight line. The INL parameter is used to set the maximum value of INLs.

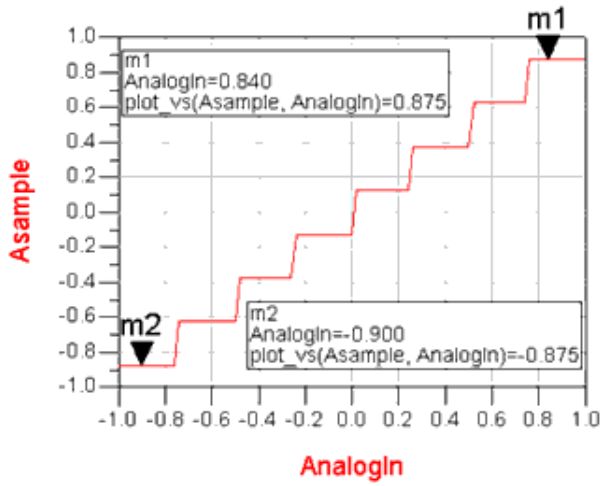
> **Note**
> There is no guarantee that the value of the INL parameter will be reached. It depends on the NBits, INL, and DNL parameter values. With larger NBits, smaller INL, and larger DNL, the INL will be reached more easily and frequently.

9. For an ideal ADC (INL=0 and DNL=0) the conversion threshold is {-VRef + i × LSB}, where i = 1, ... , $2^{NBits} - 1$.
10. For an ideal ADC (INL=0 and DNL=0) the A_out output is in the range of [-(VRef - 0.5 LSB), (VRef - 0.5 LSB)] with value {-VRef + (i - 0.5) × LSB}, where i = 1, ... , 2^{NBits}.
11. DigitalOut vs. A_in and A_out vs. A_in show the output of an ideal ADC (INL=0 and DNL=0) with NBits=3, VRef=1, and *OutputDigitalFormat* = *OffsetBinary*.
    In this case, LSB = 0.25; conversion threshold is {-0.75, -0.5, -0.25, 0.0, 0.25, 0.5, 0.75}; and A_out is {-0.875, -0.625, -0.375, -0.125, 0.125, 0.375, 0.625, 0.875}.

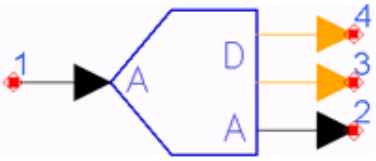
DigitalOut vs. A_in



A_out vs. A_in



AtoD_ADI



Description: Analog to Digital Converter with Input Sample and Hold
Categories: Analog/RF

Model Parameters

Name	Description	Default	Units	Type	Range
Model	A-to-D model: User specified model, AD6645_80.adc, AD6645_105.adc, AD9214_65_2V.adc, AD9214_80_1V.adc, AD9214_105_1V.adc, AD9215_65.adc, AD9215_80.adc, AD9215_105.adc, AD9216_65.adc, AD9216_80.adc, AD9216_105.adc, AD9218_40_1V.adc, AD9218_65_2V.adc, AD9218_80_1V.adc, AD9218_105_1V.adc, AD9219_40.adc, AD9219_65.adc, AD9222_40.adc, AD9222_50.adc, AD9226_2V.adc, AD9228_40.adc, AD9228_65.adc, AD9229_50.adc, AD9229_65.adc, AD9230_170.adc, AD9230_210.adc, AD9230_250.adc, AD9233_105.adc, AD9233_125.adc, AD9236.adc, AD9237_20.adc, AD9237_40.adc, AD9237_65.adc, AD9238_20.adc, AD9238_40.adc, AD9238_65.adc, AD9244_40.adc, AD9244_65.adc, AD9245_20.adc, AD9245_40.adc, AD9245_65.adc, AD9245_80.adc, AD9246_105.adc, AD9246_125.adc, AD9248_20.adc, AD9248_40.adc, AD9248_65.adc, AD9252.adc, AD9254.adc, AD9259.adc, AD9287.adc, AD9289.adc, AD9430_170_LVDS.adc, AD9430_210_LVDS.adc, AD9433_105.adc, AD9433_125.adc, AD9444.adc, AD9445_105_2V.adc, AD9445_105_3p2V.adc, AD9445_125_2V.adc, AD9445_125_3p2V.adc, AD9446_80_2V.adc, AD9446_80_3p2V.adc, AD9446_100_2V.adc,	User specified model		Enumeration	

	AD9446_100_3p2V.adc, AD9460_80_3p4V.adc, AD9460_105_3p4V.adc, AD9461_125_3p4V.adc, AD9461_130_3p4V.adc, AD9480.adc, AD9600.adc, AD9601.adc, AD9626_170.adc, AD9626_210.adc, AD9626_250.adc, AD9627-11.adc, AD9627.adc, AD9640.adc, AD80141.adc, Ideal_8_Bit.adc, Ideal_10_Bit.adc, Ideal_12_Bit.adc, Ideal_14_Bit.adc				
ModelDir	Directory for model adc files; blank means default dir			Filename	
UserModel	User specified A-to-D model			Filename	
UserNBits	User specified number of bits in the A-to-D model	8		Integer	[2:16]
UserMinSR	User specified minimum sample rate (encode rate)	0	Hz	Float	[0:∞)
UserMaxSR	User specified maximum sample rate (encode rate)	1e12	Hz	Float	(UserSR_Min:∞)
UserCommonModeOffset	User specified common mode offset voltage applied to input signal	0	V	Float	[0:∞)
UserInputSpan	User specified input signal span	2	V	Float	(0:∞)
UserDigitalFormat	User specified output digital format: Offset binary, Twos-complement	Twos-complement		Enumeration	
CenterFreq	Spectral center frequency of analog input (used when input is real and NyquistZone is zero)	0	Hz	Float	[0:∞)
NyquistZone	Positive value representing the analog input Nyquist zone	1		Integer	[0:∞)
EnableExtJitter	Enable external jitter: NO, YES	NO		Enumeration	
ExtJitter	External RMS jitter applied to the input signal	0	s	Float	[0:∞)

Input Ports

Port	Name	Description	Signal Type	Optional
1	A_in	input analog signal	envelope	NO

Output Ports

Port	Name	Description	Signal Type	Optional
2	A_out	output sampled analog baseband signal	envelope	NO
3	D_I	output NBit word as integer for I channel	int	NO
4	D_Q	output NBit word as integer for Q channel	int	NO

Notes/Equations

AtoD_ADI component is only supported ion 32-bit Windows.

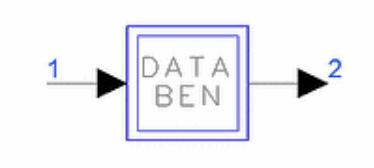
1. This component models analog-to-digital converters from Analog Devices with

reference to Analog Devices part numbers. The input at A_in can be a real baseband signal or a complex envelope signal. The output digital words are in integer form. When the input is a real baseband signal, the output digital word is at the D_I output. When the input is a complex envelope signal, the D_I output contains the digital word for the I envelope and the D_Q ouput contains the digital word for the Q envelope. The A_out contains the quantized form of the input signal. If A_in is a complex envelope signal, then the A_out is also a complex envelope signal at the same characterization frequency.

2. This block reads 1 sample from the input and writes 1 sample to the outputs.
3. ModelDir is the computer disk directory location in which the Analog Devices AtoD model files (*.adc) are located. The default location is in the ADS installation directory under adsptolemy\lib\models\ADI. The user can use these model files delivered with ADS or use other Analog Devices AtoD model files obtained from Analog Devices and located in another directory.
4. When Model is not 'User specified model'
   The *Model* value is the name of an Analog Devices AtoD model file that is delivered with ADS. By selecting one of these predefined models, the full characteristics are defined and used. For these models, the simulator prints out the values for NBits, MinSR, MaxSR, CommonModeOffset, InputSpan and DigitalFormat associated with the selected Analog Devices AtoD model.
5. When Model is 'User specified model'
   The *UserModel* is the Analog Devices *.adc filename that the user separately obtained and would like to use. This option is available when the predefined *.adc filenames are not what the user wants to use and would like to use instead other Analog Devices AtoD models. The *UserNBits*, *UserMinSR*, *UserMaxSR*, *UserCommonModeOffset*, *UserInputSpan*, and *UserDigitalFormat* values are to be entered by the user as is defined for the *UserModel* listed by the user and are value to be obtained from Analog Devices for the model listed.
6. Every simulation sample results in sampling the input A_in and quantization of that input resulting in the output values. When *EnableExtJitter* = YES, then the *ExtJitter* defines the RMS jitter applied to the input A_in before sampling.
7. The Analog Devices AtoD (*.adc) model is defined for use with the input signals in the range [CommonModeOffset, CommonModeOffset + InputSpan]. The CommonModeOffset bias is automatically applied internally to the AtoD_ADI input A_in. Thus, the AtoD_ADI input A_in is defined for use with A_in in the range [-InputSpan / 2, InputSpan / 2), and limited to these limits when A_in is outside this range. The CommonModeOffset and InputSpan are automatically defined when Model is not 'User specified model'. They are defined by *UserCommonModeOffset* and *UserInputSpan* when Model is 'User specified model'.
8. Quantization of the input A_in
   NBits, InputSpan and DigitalFormat are used to quantize the input A_in to generate the digital words that appear at the integer outputs D_I and D_Q and the quantized analog output A_out. These values are either predefined when Model is not 'User specified model', or are user specified when Model is 'User specified model'. Define VRef = InputSpan/2. Define LSB = least significant bit = $2 \times VRef / 2^{NBits}$.
   The conversion thresholds for A_in are {-VRef + i × LSB}, where i = 1, ... , $2^{NBits} - 1$.
   When DigitalFormat is 'Offset binary', the digital words span integer values in the range [0, 2^ NBits - 1].
   When DigitalFormat is 'Twos-complement', the digital words span integer values in the range [-2^(NBits /2), 2^(NBits /2) - 1].
   A_out output is in the range of [-(VRef - 0.5 LSB), (VRef - 0.5 LSB)] with values {-VRef + (i - 0.5) × LSB}, where i = 1, ... , 2^{NBits} .

9. The Analog Devices AtoD models all have a start up transient time (Latency) during which time the output values are random values.
10. The Analog Devices AtoD models allow the user to specify the spectral center frequency of the analog signal (CenterFreq, when A_in is real) and the analog input Nyquist zone to improve AtoD model behavior.
11. See the Analog Devices AtoD documentation for more detail on their AtoD models. http://www.analog.com/en/analog-to-digital-converters/ad-converters/products/index.html

BinaryCoder



Description: Binary data coder
Library: Timed, Data Processing
Class: TSDF_BinaryCoder
Derived From: baseOmniSysStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
SymbolTime	input data symbol time	0.001	sec	real	[TStep, ∞)†
Type	differential data coding type: Differential data encoder, Differential data decoder	Differential data encoder		enum	

† TStep is the simulation time step for the component input signal.

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
2	output	output signal	timed

Notes/Equations

1. This component differentially encodes (for Type=Differential data encoding) or decodes (for Type=Differential data decoding the input binary signal. Input data is assumed to be in NRZ format and output data is in NRZ format.
   For Type=Differential data encoding, the following equations describe the encoding process.
   The input NRZ data signal $V_1(t)$ is first converted into a logic signal $V_{L1}(t)$:

   $$V_{L1(t)} = \begin{cases} 1 & \text{if } V_1(t) \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

   The output logic state voltage VL2(t) is determined by:

$V_{L2}(t) = V_{L1}(t) \oplus V_{L2}(t \; SymbolTime)$

where $\oplus$ denotes the EXCLUSIVE-OR operation.
The output signal $V_2(t)$ is then calculated:

$$V_2(t) = \begin{cases} +1 & \text{if } V_{L2}(t) = 1 \\ -1 & \text{if } V_{L2}(t) = 0 \end{cases}$$

For Type=Differential data decoding, the following equations describe the decoding process.
The input NRZ data signal $V_1(t)$ is first converted into a logic signal $V_{L1}(t)$:

$$V_{L1(t)} = \begin{cases} 1 & \text{if } V_1(t) \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

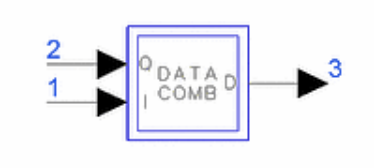
Let
$V_A = (V_{L1}(t) + V_{L1}(t \text{ SymbolTime}))$ modulo 2.

The output signal $V_2(t)$ is then calculated:

$$V_2(t) = \begin{cases} +1 & \text{if } \text{abs}(-V_A(t)) > 0.5 \\ -1 & \text{if } \text{abs}(-V_A(t)) \leq 0.5 \end{cases}$$

2. For information regarding timed data processing component signals, refer to the *Timed Data Processing Components* (timed).

BinaryCombiner



Description: Binary data combiner
Library: Timed, Data Processing
Class: TSDF_BinaryCombiner
Derived From: baseOmniSysStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
SymbolTime	input data symbol time	0.001	sec	real	[2 * TStep, ∞)†
Delay	data input time delay from t=0 for start of data stream	0	sec	real	{-1} or [0, ∞)††

† TStep is the simulation time step for the component input signals.
†† Refer to Note 2 for the special meaning of a -1 value.

Pin Inputs

Pin	Name	Description	Signal Type
1	I_in	I input	timed
2	Q_in	Q input	timed

Pin Outputs

Pin	Name	Description	Signal Type
3	output	output signal	timed

Notes/Equations

1. The input data signals are presumed to be in NRZ binary format with logic 0 level of -1V and logic l level of +1V.
2. Signal Transformation Model
   The output data stream is in the binary NRZ format with +1 and -1 levels and with a symbol time of SymbolTime/2.
   The data combiner performs an integrate and dump operation on the input data to determine the state of the input data at pins 1 and 2. The value of Delay is used to set the time instant at which the integration begins. If the value of the integral is positive at the end of the integration period, then a decision is made that the input bit is a +1, else the decision is that the input data is a -1. The integration period is

equal to SymbolTime.

When Delay = -1, self synchronization is performed. The time at which either input's absolute voltage is greater than 0.5V is determined. This time instant is the time when the integrate and dump process is first started. For the time previous to this instant, the output is set equal to 0.

The output data stream at pin 3 combines input data at pins 1 and 2 by interleaving alternate data bits; the first data bit out is the first data bit at pin 1.

Let

$$T_i = \text{start of an integrate and dump interval}$$
$$= \text{Delay} + \text{i SymbolTime}; \; i = 0, 1, 2$$

Let

$$\text{SUM1} = \int_{T_i}^{T_{i+1}} V_1(t)dt$$

Let

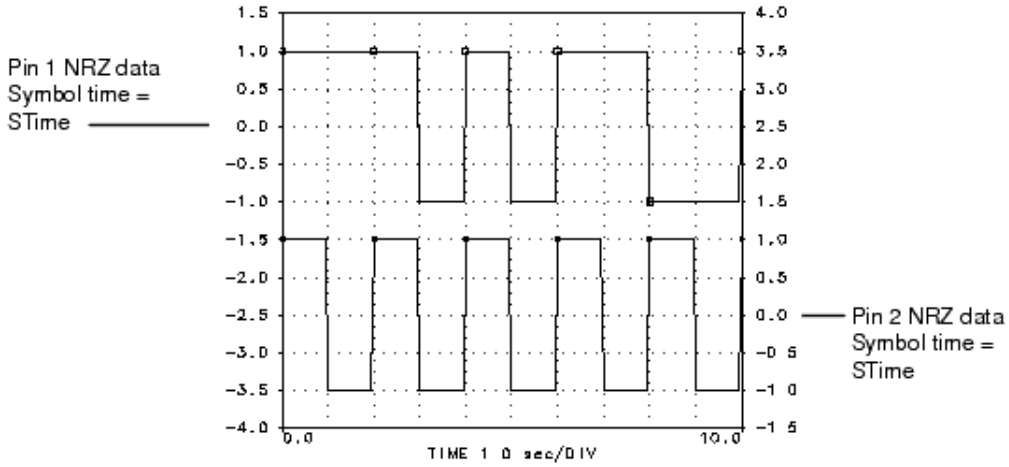
$$\text{SUM2} = \int_{T_i}^{T_{i+1}} V_2(t)dt$$

Then

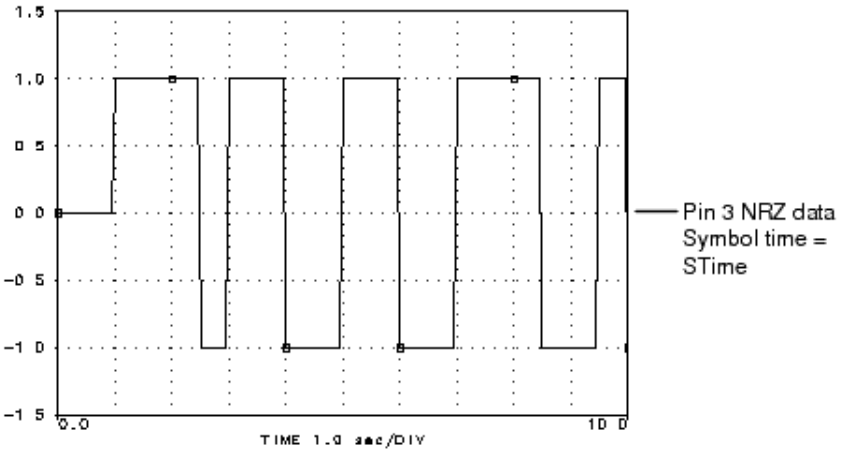
$$V_{3i}(t) = \begin{cases} 0 & \text{for t} < \text{T}_1 \\ -1; & if \text{ SUM1} < 0; \quad \text{for T}_i + SymbolTime \le t \le T_i + 3SymbolTime/2 \\ 1; & if \text{ SUM1} \ge 0; \quad for \; T_i + SymbolTime \le t \le T_i + 3SymbolTime/2 \\ -1; & if \text{ SUM2} < 0; \quad for \; \text{T}_i + (3SymbolTime/2) \le t \le T_i + 2SymbolTime \\ 1; & if \; SUM2 \ge 0; \quad for \; \text{T}_i + (3SymbolTime/2) \le t \le T_i + 2SymbolTime \end{cases}$$

The output bits alternate from pin 1 and 2, as shown in BinaryCombiner Inputs and BinaryCombiner Output. Note that the output signal is causal-therefore the output is equal to 0V until after the synchronization time.

BinaryCombiner Inputs

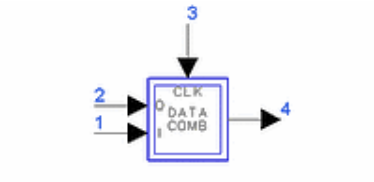


BinaryCombiner Output



3. For information regarding timed data processing component signals, refer to *Timed Data Processing Components* (timed).

BinaryCombinerClocked



Description: Binary data combiner with clock input
Library: Timed, Data Processing
Class: TSDF_BinaryCombinerClocked
Derived From: baseOmniSysStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	I_in	I input	timed
2	Q_in	Q input	timed
3	clock	clock signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
4	output	output signal	timed

Notes/Equations

1. The input data signals are presumed to be in NRZ binary format with logic 0 level of −1V and logic l level of +1V. The output is also in NRZ binary format.
2. This component converts two input NRZ data streams into a single output data stream by interleaving the two inputs. Pins 1 and 2 are for the two input data streams and the third is for an input clock. At every alternate positive edge of the input clock, the two input signals are sampled, compared to a threshold of 0V and decoded as a logical 0 or 1.
   Output voltage $V_4(t)$ is set to 0 before the first positive edge of the input clock.

   The same input clock is also used to clock the output data. The data of the first input pin is output at the alternate positive clock edges at which the input pins are sampled; data of the second input pin is output at the other positive clock edges. The input clock period should equal the output data symbol time and the output data symbol time is half the input symbol time.
   Mathematically, this component can be described as:
   $V_1(t)$ and $V_2(t)$ are the input data voltages

   $V_3(t)$ is the input clock; $V_4(t)$ is the output signal

$V_1(t)$, $V_2(t)$, and $V_3(t)$ are assumed to be NRZ signals with −1 and +1 voltage states

Let T_0, T_1, T_2, ... be the time instances when the positive edges of the input clock $V_3(t)$ occur (a positive edge occurs at the instant when the clock voltage changes from a negative to a non-negative value).

For all k = 0,1,2, ... let

$$A_k = \begin{cases} 1 & if\ V_1(T_{2K}) \geq 0 \\ -1 & if\ V_1(T_{2K}) < 0 \end{cases}$$

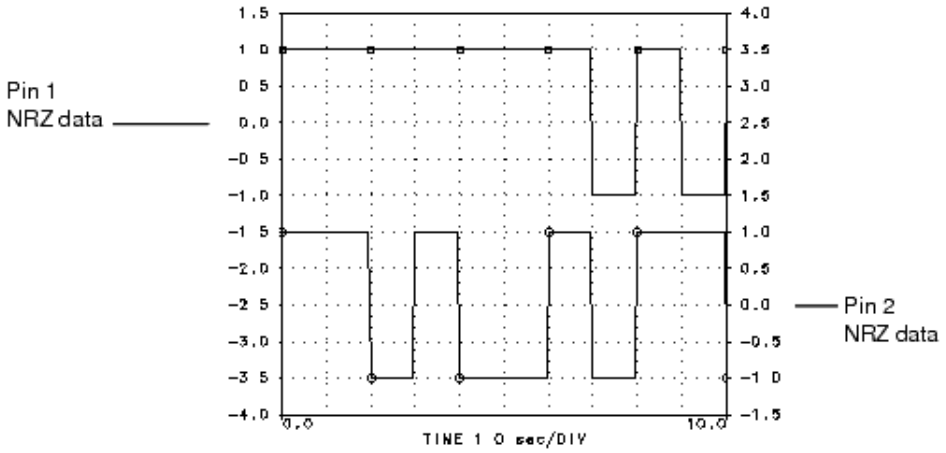
$$B_k = \begin{cases} 1 & if\ V_2(T_{2K}) \geq 0 \\ -1 & if\ V_2(T_{2K}) < 0 \end{cases}$$

then

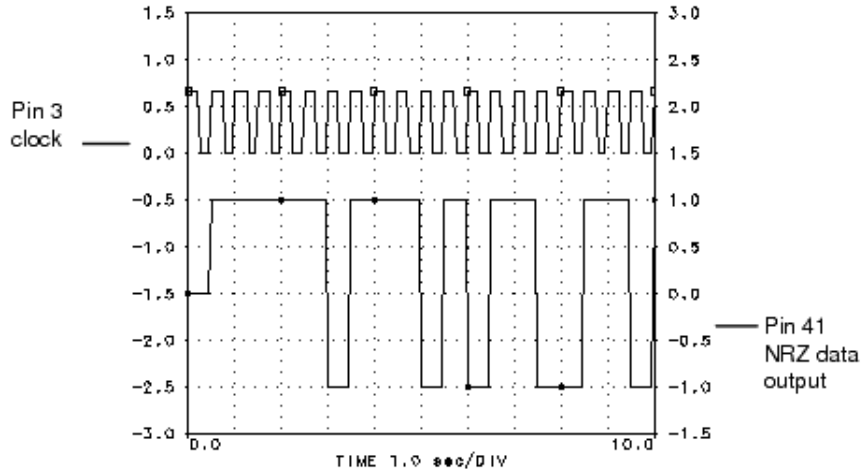
$$V_4(t) = \begin{cases} A_K & for\ T_{2K} \leq t < T_{2K+1} \\ B_K & for\ T_{2K+1} \leq t < T_{2K+2} \end{cases}$$

3. The relationship between the input and output signals is shown in BinaryCombinerClocked Inputs and BinaryCombinerClocked Clock and Output.

BinaryCombinerClocked Inputs

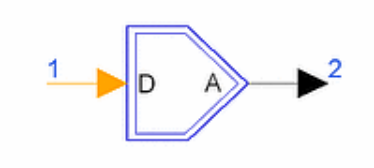


BinaryCombinerClocked Clock and Output



4. For information regarding timed data processing component signals, refer to *Timed Data Processing Components* (timed).

DAC_Timed



Description: Digital-to-analog converter with integral and differential nonlinearities
Library: Timed, Data Processing
Class: TSDF_DAC_Timed

Parameters

Name	Description	Default	Unit	Type	Range
ROut	resistance	0.0	Ohm	real	[0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
TStep	output time step; TStep=0 results in use of externally set TStep	0.0	sec	real	[0, ∞)
NBits	number of bits	8		int	[2, ∞)
VRef	reference voltage for output analog signal A: -VRef<=A<=VRef	1.0	V	real	(0, ∞)
INL	integral nonlinearity relative to least significant bit (LSB)	0.0		real	[DNL/2, ∞)
DNL	differential nonlinearity relative to least significant bit (LSB)	0.0		real	[0, ∞)
UpSampleRatio	output upsampling ratio	1		int	[1, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	D	input to D/A	int

Pin Outputs

Pin	Name	Description	Signal Type
2	A	output of D/A	timed

Notes/Equations

1. This component models a digital-to-analog converter with integral and differential nonlinearities. The input is a digital word in integer form. The output is baseband analog signal.

   Note
   Bit-to-integer conversion is not performed within this model; for this conversion, place an external BitsToInt converter before this component.

2. To avoid clipping, the input integer D must be in the range of [0, $2^{NBits} - 1$].
3. DNL (differential nonlinearity) error is defined as the difference between an actual output step width and the ideal value of 1 LSB (least significant bit, 1 LSB = 2 × VRef / 2^{NBits}). For an ideal digital-to-analog converter, in which the DNL=0 LSB, each

output analog step equals 1 LSB. The DNL parameter is used to set the maximum value of DNLs. A DNL error specification of less than or equal to 1 LSB guarantees a monotonic transfer function.

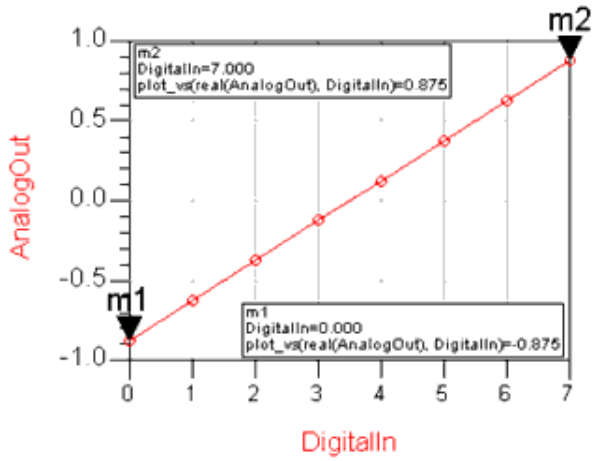
> **Note**
> There is no guarantee that the DNL parameter value will be reached. The DNL error is modeled by a normal (Gaussian) distribution. The distribution has an approximate 1% probability that the DNL error will be equal to or greater than the DNL parameter value (or less than -DNL). Those numbers are then truncated to DNL (or -DNL) before further processing.

4. INL (integral nonlinearity) error is defined as the deviation (in LSB) of the digital-to-analog converter actual transfer function from an ideal straight line. The INL parameter is used to set the maximum value of INLs.

> **Note**
> There is no guarantee that the INL parameter value will be reached; it depends on the NBits, INL, and DNL values. With greater NBits, smaller INL, and greater DNL, the INL will be reached more easily and frequently.

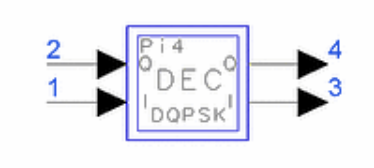
5. For an ideal DAC (INL=0 and DNL=0) the output signal A is in the range of [-(VRef - 0.5 LSB), (VRef - 0.5 LSB)] with value {-VRef + (i - 0.5) × LSB}, where i=1, ... , 2^{NBits}.
6. For every input word, DAC_Timed will output a number of output samples equal to UpSampleRatio.
7. AnalogOut vs. DigitalIn shows the output of an ideal DAC (INL=0 and DNL=0) with NBits=3, VRef=1, and UpSampleRatio=1.
   In this case, the LSB = 0.25; and the output signal A is {-0.875, -0.625, -0.375, -0.125, 0.125, 0.375, 0.625, 0.875}.

AnalogOut vs. DigitalIn



8. For information regarding timed data processing component signals, refer to the *Introduction* (timed).

DQPSK_Pi4Decoder



Description: pi/4-DQPSK IQ data decoder
Library: Timed, Data Processing
Class: TSDFDQPSK_Pi4Decoder
Derived From: baseSubcircuit

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
SymbolTime	input data symbol time	0.001	sec	real	[newpro:TStep, ∞)†

† TStep is the simulation time step for the component input signals.

Pin Inputs

Pin	Name	Description	Signal Type
1	I_in	I input	timed
2	Q_in	Q input	timed

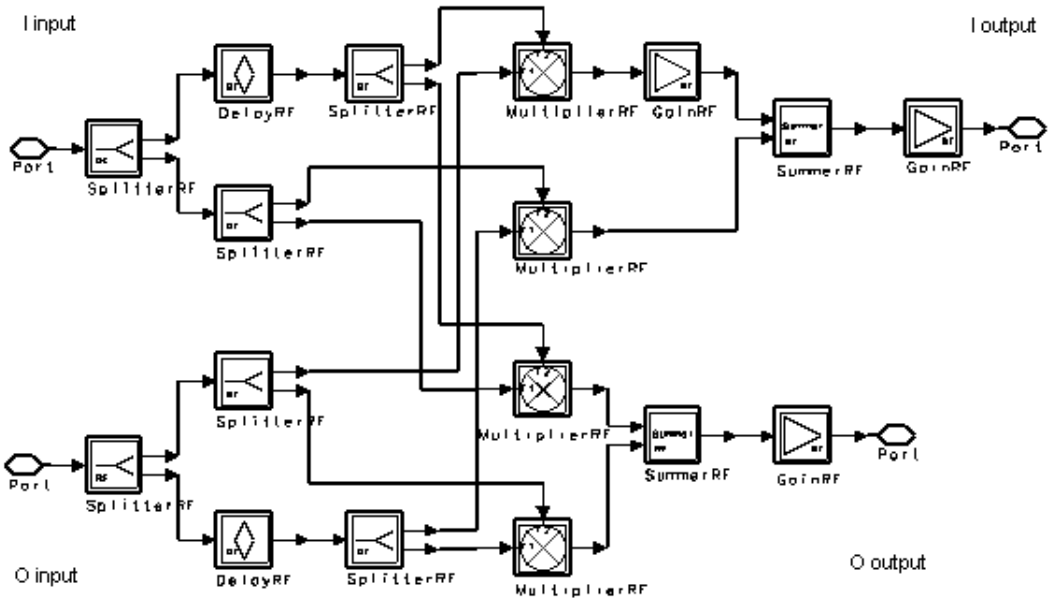
Pin Outputs

Pin	Name	Description	Signal Type
3	I_out	I output	timed
4	Q_out	Q output	timed

Notes/Equations

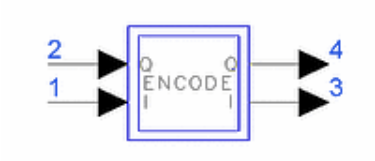
1. This component is composed of other components. DQPSK Pi/4 Data Decoder Network shows the network used to implement the decoder. This component is used when the differential encoding is performed prior to modulation and the modulated signal is coherently demodulated. It is assumed that differential encoding is performed with the EncoderIQ component with the parameter Type=pi4dqpsk. The output of the coherent demodulator should be fed directly to the DQPSK_Pi4Decoder component to perform differential decoding (refer to documentation for the DQPSK_Pi4DemodSync coherent component).

DQPSK Pi/4 Data Decoder Network



2. For information regarding timed data processing component signals, refer to the *Timed Data Processing Components* (timed).

EncoderIQ



Description: Encoder for IQ data
Library: Timed, Data Processing
Class: TSDF_EncoderIQ
Derived From: baseOmniSysStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
SymbolTime	input data symbol time	0.001	sec	real	[TStep, ∞)†
Delay	time delay from t=0 for start of integrate and dump detection of input data stream, Delay = -1 for auto synchronization	0	sec	real	{-1} or [0, ∞)††
Type	type of IQ data encoder: Uncoded, DQPSK, PI/4 DQPSK	Uncoded		enum	

† TStep is the simulation time step for the component input signals.
†† Refer to Note 2 for the special meaning of a -1 value.

Pin Inputs

Pin	Name	Description	Signal Type
1	I_in	I input	timed
2	Q_in	Q input	timed

Pin Outputs

Pin	Name	Description	Signal Type
3	I_out	I output	timed
4	Q_out	Q output	timed

Notes/Equations

1. The input data signals are presumed to be in the NRZ binary format, with logic 0 level at -1V and logic 1 level at +1V.
2. This component can be used to differentially encode data according to either the DQPSK or the PI/4 DQPSK formats (also refer to component *EncoderIQClocked* (timed)).
   Data encoding begins after the delay time (Delay) has elapsed. Inputs are integrated

beginning at t=Delay and dumped at SymbolTime intervals. The I and Q output states are then calculated from the results of the integrate and dump operations. Due to this integrate and dump operation, the component introduces a delay of SymbolTime at its output.

When Delay=-1, self-synchronization is performed by determining the time at which any input's absolute voltage is > 0.5V. This time instant is the time when the integrate and dump process begins. For the time previous to this instant, the outputs are set.

Initial conditions:

Uncoded: $V_3(t) = V_4(t) = 0$ for $t < T_0$

DQPSK: $V_3(t) = \cos(3\pi/4)$, $V_4(t) = \sin(3\pi/4)$ for $t < T_0$

PI/4 DQPSK: $V_3(t) = 1$, $V_4(t) = 0$ for $t < T_0$

3. The I channel input $V_1(t)$ is at pin 1; the Q channel input $V_2(t)$ is at pin 2.

   Let X_K and Y_K denote the logic states of the I and Q input.

$$X_K = \begin{cases} 1 & if \int_{T_D+kT_S}^{T_D+(k+1)T_S} V_1(t)\,dt \geq 0 \\ 0 & otherwise \end{cases}$$

$$Y_K = \begin{cases} 1 & if \int_{T_D+kT_S}^{T_D+(k+1)T_S} V_2(t)\,dt \geq 0 \\ 0 & otherwise \end{cases}$$

   Let I_k and Q_k denote the corresponding differentially encoded data.

   The corresponding voltage levels of the I and Q outputs I_K and Q_K are calculated depending on Type.

   For Type=Uncoded. No data encoding is done on the inputs and the I and Q outputs are set equal to the I and Q inputs, respectively.

   Therefore,

   $V_3(t) = V_1(t)$ and $V_4(t) = V_2(t)$

   For Type=DQPSK. Differential encoding is performed:

   $I_0 = \cos(3\pi/4)$ $Q_0 = \sin(3\pi/4)$

   $I_K = I_{k-1} \cos(\Delta) - Q_{k-1} \sin(\Delta)$, $k > 0$

   $Q_K = I_{k-1} \sin(\Delta) + Q_{k-1} \cos(\Delta)$, $k > 0$

   where

$$\Delta\phi = \begin{cases} \pi & \text{for } X_K = 1, Y_K = 1 \\ \pi/2 & \text{for } X_K = 0, Y_K = 1 \\ 0 & \text{for } X_K = 0, Y_K = 0 \\ -\pi/2 & \text{for } X_K = 1, Y_K = 0 \end{cases}$$

   For Type=PI/4 DQPSK. Differential encoding is performed:

   $I_0 = 1$ $Q_0 = 0$

   $I_K = I_{k-1} \cos(\Delta) - Q_{k-1} \sin(\Delta)$, $k > 0$

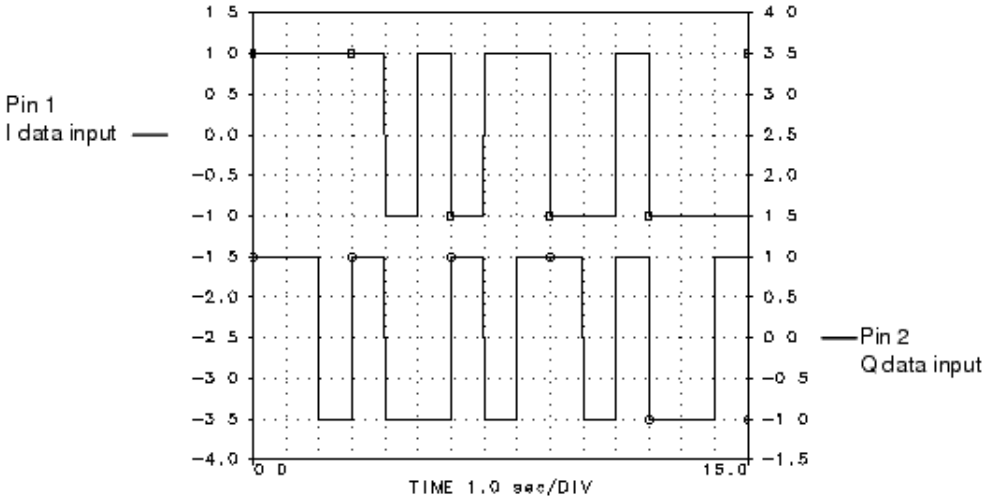
$Q_K = I_{k-1} \sin(\Delta) + Q_{k-1} \cos(\Delta), k > 0$

where

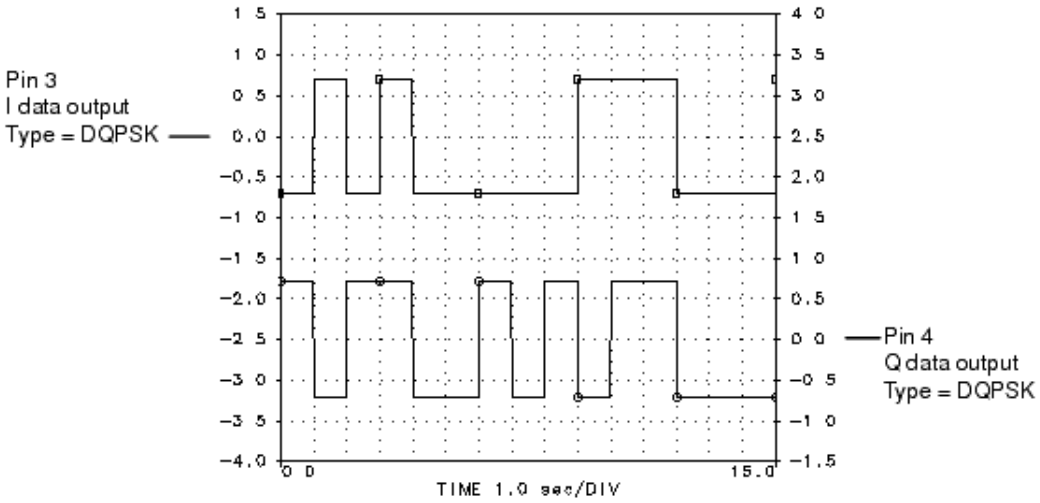
$$\Delta\phi = \begin{cases} -3\pi/4 & \text{for } X_K = 1, Y_K = 1 \\ 3\pi/4 & \text{for } X_K = 0, Y_K = 1 \\ \pi/4 & \text{for } X_K = 0, Y_K = 0 \\ -\pi/4 & \text{for } X_K = 1, Y_K = 0 \end{cases}$$

4. Output signal is causal; therefore, the output is equal to the 0V initial condition until after synchronization time.
5. I and Q Data Inputs through I and Q Data Outputs, Type = PI/4DQPSK depict the input and various outputs of the EncoderIQ component.

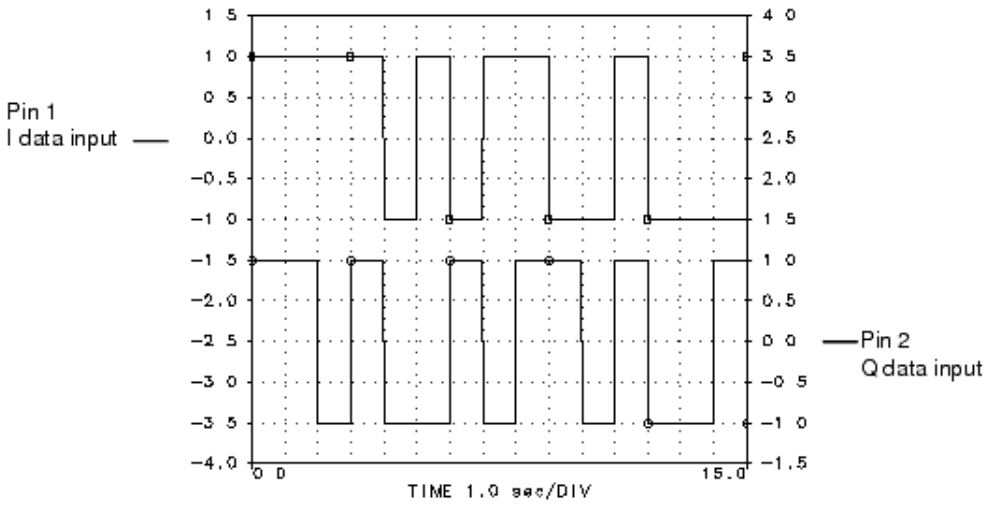
I and Q Data Inputs



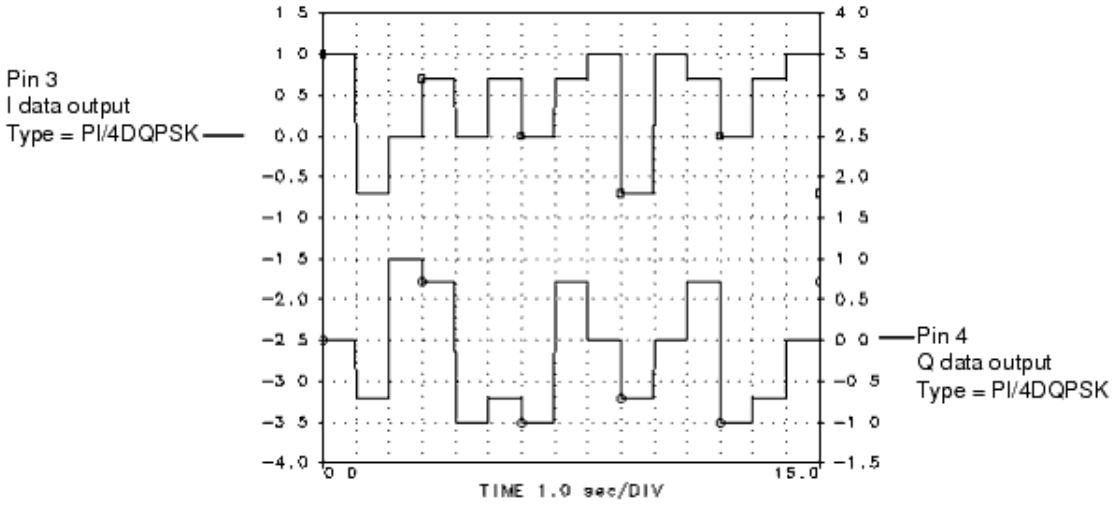
I and Q Data Outputs, Type = DQPSK



I and Q Data Inputs

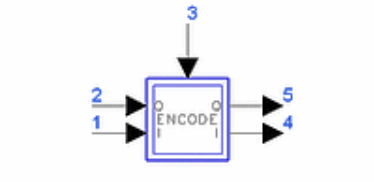


I and Q Data Outputs, Type = PI/4DQPSK



6. For information regarding timed data processing component signals, refer to *Timed Data Processing Components* (timed).

EncoderIQClocked



Description: Encoder for IQ Data with clock input
Library: Timed, Data Processing
Class: TSDF_EncoderIQClocked
Derived From: baseOmniSysStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
Type	type of IQ data encoder: Uncoded, DQPSK, PI/4 DQPSK	Uncoded		enum	

Pin Inputs

Pin	Name	Description	Signal Type
1	I_in	I input	timed
2	Q_in	Q input	timed
3	clock	clock signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
4	I_out	I output	timed
5	Q_out	Q output	timed

Notes/Equations

1. The input data signals are presumed to be in the NRZ binary format, with logic 0 level at −1V and logic 1 level at +1V.
2. This component can be used to differentially encode data according to either the DQPSK format or the PI/4 DQPSK format.
   I channel input $V_1(t)$ is at pin 1;

   Q channel input $V_2(t)$ is at pin 2;

   clock input $V_3(t)$ is at pin 3;

   encoded I channel output $V_4(t)$ is at pin 4;

   encoded Q channel output $V_5(t)$ is at pin 5.

   Let T_0, T_1, T_2, ... be the time instances when the positive edges of the input clock $V_3(t)$ occur (a positive edge occurs at the instant when the clock voltage

changes from a negative to a non-negative value). At each positive clock edge, I and Q inputs are sampled and compared to a 0V threshold to determine if inputs are at a logic 0 or 1 state; I and Q outputs are derived from these states.
Initial conditions:
Uncoded: $V_4(t) = V_5(t) = 0$ for $t < T_0$

DQPSK:
$V_4(t) = \cos(3\pi/4)$ for $t < T_0$

$V_5(t) = \sin(3\pi/4)$ for $t < T_0$

PI/4 DQPSK: $V_4(t) = 1$ and $V_5(t) = 0$ for $t < T_0$

Let X_K and Y_K denote the Boolean logic states of the I and Q input at the kth sampling instance.

$$X_K = \begin{cases} 1 & if\ V_1(T_K) \geq 0.5 \\ 0 & otherwise \end{cases}$$

$$Y_K = \begin{cases} 1 & if\ V_2(T_K) \geq 0.5 \\ 0 & otherwise \end{cases}$$

Let I_K and Q_K denote the differentially encoded data.

For Type=Uncoded No data encoding is done on the inputs and the I and Q outputs are set equal to the I and Q inputs, respectively.
Therefore
$V_4(t) = V_1(t)$ $V_5(t) = V_2(t)$

For Type=DQPSK
Initial conditions:
$I_0 = \cos(3\pi/4)$

$Q_0 = \sin(3\pi/4)$

Then
$I_K = I_{k-1}\cos(\Delta\varphi) - Q_{k-1}\sin(\Delta\varphi)$, $k > 0$

$Q_K = I_{k-1}\sin(\Delta\varphi) + Q_{k-1}\cos(\Delta\varphi)$, $k > 0$

where

$$\Delta\phi = \begin{cases} \pi & \text{if } X_K = 1, Y_K = 1 \\ \pi/2 & if\ X_K = 0, Y_K = 1 \\ 0 & if\ X_K = 0, Y_K = 0 \\ -\pi/2 & if\ X_K = 1, Y_K = 0 \end{cases}$$

For Type=PI/4 DQPSK
Initial conditions:
$I_0 = 1$

$Q_0 = 0$

Then
$I_K = I_{k-1}\cos(\Delta\varphi) - Q_{k-1}\sin(\Delta\varphi)$, $k > 0$

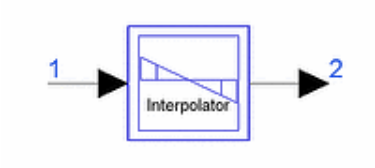
$Q_K = I_{k-1}\sin(\Delta\varphi) + Q_{k-1}\cos(\Delta\varphi)$, $k > 0$

where

$$\Delta\phi = \begin{cases} -3\pi/4 & \text{if } X_K = 1, Y_K = 1 \\ 3\pi/4 & \text{if } X_K = 0, Y_K = 1 \\ \pi/4 & \text{if } X_K = 0, Y_K = 0 \\ -\pi/4 & \text{if } X_K = 1, Y_K = 0 \end{cases}$$

3. For information regarding timed data processing component signals, refer to *Timed Data Processing Components* (timed).

Interpolator



Description: Interpolator for timed signals
Library: Timed, Data Processing
Class: TSDF_Interpolator
Derived From: baseOmniSysStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
TimeRef	reference time for interpolated data; will be normalized by modulo tstep to a value, tref, in the range to 0 <= tref <= tstep	0	sec	real	[0, TStep]†
Type	type of interpolator: none, linear, cubic spline, n-point Lagrange	none		enum	
Order	order of Lagrange interpolator	4		int	††

† TStep is the simulation time step for the component input signal.
†† Refer to Note 1 regarding Order parameter values.

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
2	output	output signal	timed

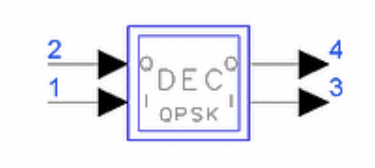
Notes/Equations

1. This interpolator implements linear, cubic spline, or Lagrange interpolation. The orders of interpolation are:
   linear: 2
   cubic spline: 4
   n-point Lagrange: where n ≥ 4
   When Type = linear, this component introduces a delay of TStep − TimeRef.
   When Type = cubic spline, this component introduces a delay of 2 × TStep − TimeRef.
   When Type is n-point Lagrange, this component inroduces a delay of
   (Order/2) × TStep − TimeRef, when Order is even
   ((Order −1)/2) × TStep − TimeRef, when Order is odd

For linear and cubic spline interpolation, changing the Order parameter will not affect the simulation as their default orders will be used. The Order parameter is used to control the n-point Lagrange interpolation.

2. For information regarding timed data processing component signals, refer to the *Timed Data Processing Components* (timed).

QPSK_Decoder



Description: QPSK IQ data decoder
Library: Timed, Data Processing
Class: TSDFQPSK_Decoder
Derived From: baseSubcircuit

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
SymbolTime	input data symbol time	0.001	sec	real	[newpro:TStep, ∞)†

† TStep is the simulation time step for the component input signals.

Pin Inputs

Pin	Name	Description	Signal Type
1	I_in	I input	timed
2	Q_in	Q input	timed

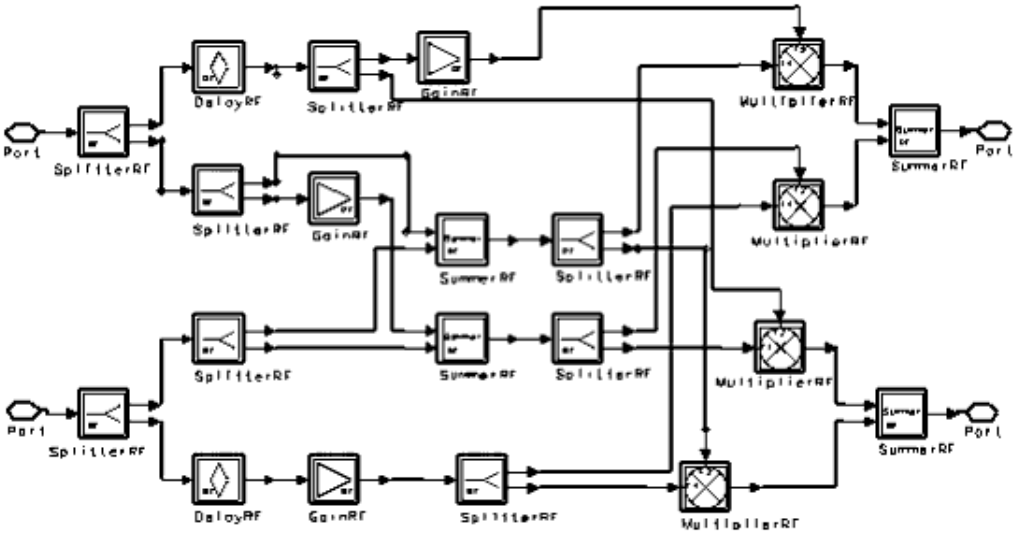
Pin Outputs

Pin	Name	Description	Signal Type
3	I_out	I output	timed
4	Q_out	Q output	timed

Notes/Equations

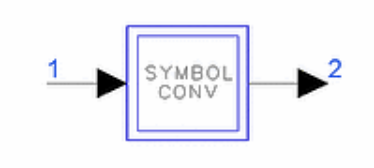
1. This component is composed of other components. QPSK Data Decoder Network shows the implementation of the network representing the QPSK differential decoder. This component is to be used when differential encoding is performed prior to modulation and the modulated signal is coherently demodulated. It is assumed that differential encoding is performed with the EncoderIQ component with Type=DQPSK (differential encoding and decoding are required because the carrier recovery circuit has a 90-degree phase ambiguity). The output of the QPSK_Demod demodulator should be fed directly to the QPSK_Decoder component to perform the differential decoding (for more information, refer to QPSK_Demod component documentation).

QPSK Data Decoder Network



2. For information regarding timed data processing component signals, refer to *Timed Data Processing Components* (timed).

SymbolConverter



Description: Symbol format converter
Library: Timed, Data Processing
Class: TSDF_SymbolConverter
Derived From: baseOmniSysStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
SymbolTime	input data symbol time	0.001	sec	real	[TStep, ∞)†
Delay	data input time delay from t=0 for start of the data stream, Delay= -1 for auto synchronization	0	sec	real	{-1} or [newpro:0, ∞)††
CodeIn	input data format: nrzIn, rzIn, cmiIn, ami_nrzIn, ami_rzIn, pam4In, pam8In, pam16In, pam32In, pam64In, pam128In, pam256In, pam512In, pam1024In, pam2048In	nrzIn		enum	†††
CodeOut	output data format: nrzOut, rzOut, cmiOut, ami_nrzOut, ami_rzOut, pam4Out, pam8Out, pam16Out, pam32Out, pam64Out, pam128Out, pam256Out, pam512Out, pam1024Out, pam2048Out	nrzOut		enum	†††

† TStep is the simulation time step for the component input signal.
†† Refer to Note 2 for the special meaning of a -1 value.
††† Either CodeIn or CodeOut must be NRZ format.

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
2	output	output signal	timed

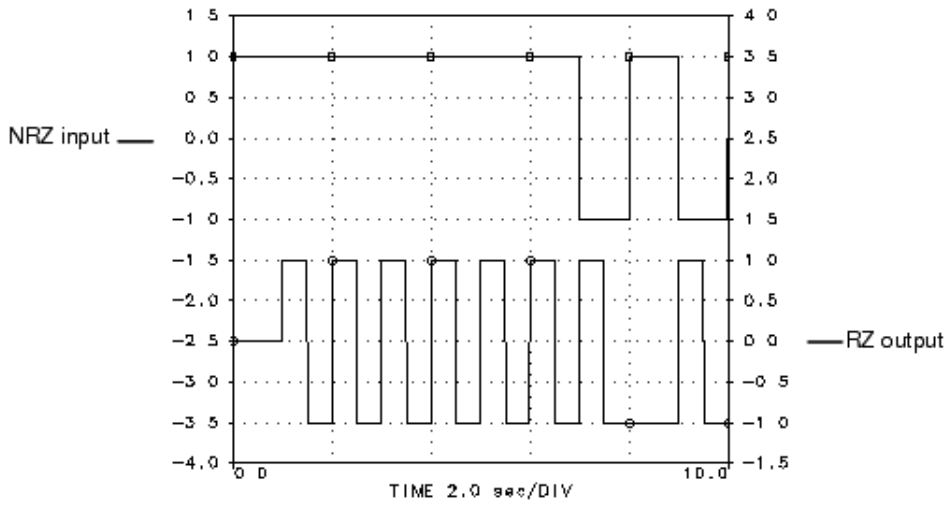
Notes/Equations

1. This component can be used to convert data in NRZ format to other formats, or it can

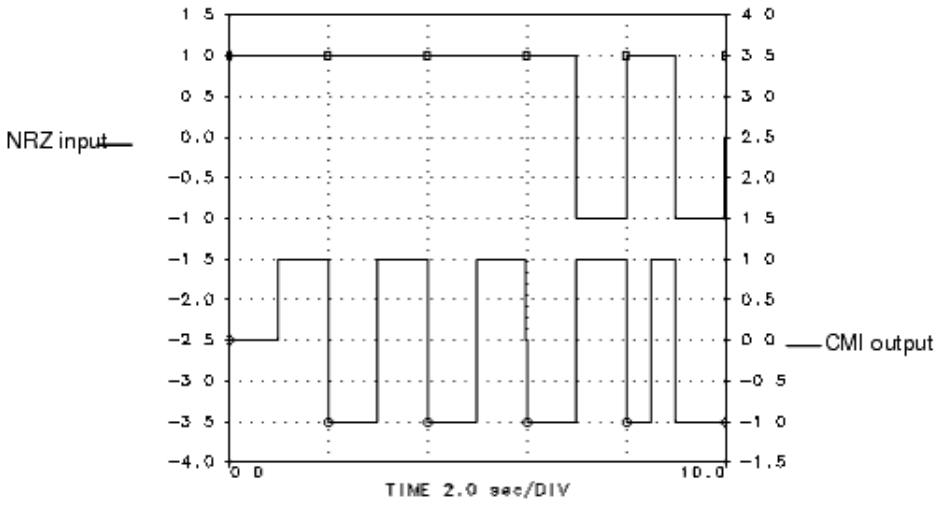
be used to convert data from other formats to NRZ format.

2. Data conversion begins after the delay time (Delay) has elapsed.
   Delay=−1 results in self-synchronization of the data combiner by detecting zero crossings. The program performs self-synchronization by determining the time at which any input's absolute voltage is greater than 0.5V.
3. The data codes are interpreted as follows:
   Binary data: nrzIn, rzIn, cmiIn. For nrz or rz, the signal has a voltage of −1 for a logic zero and +1 for a logic one; cmi (coded mark inversion), however, is a form of coding in which logic ones are represented by alternate high and low levels over the whole clock period, and logic zeros are represented by a low- to high-level transition.
   Ternary data: ami_nrzIn, ami_rzIn. The signal has a voltage of 0 for a logic zero and +1 or −1 for a logic one.
   Multi-level data: pam4In, ... , pamnIn, ... , pam2048In. The signal has a voltage of $(2i/(2^N - 1)) - 1$ for $i = 0, \ldots, 2^N - 1$.
4. NRZ Input Versus RZ Output through NRZ Input Versus PAM2048 Output demonstrate conversion from NRZ input format to each possible output format; the input symbol time is 1 µsec.
   The output signal is causal; therefore, the output is equal to the 0V initial condition until after the synchronization time.

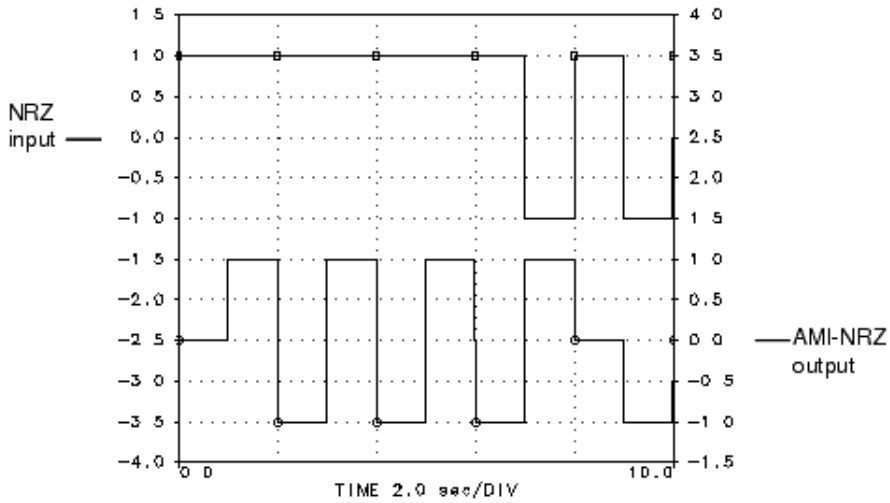
NRZ Input Versus RZ Output



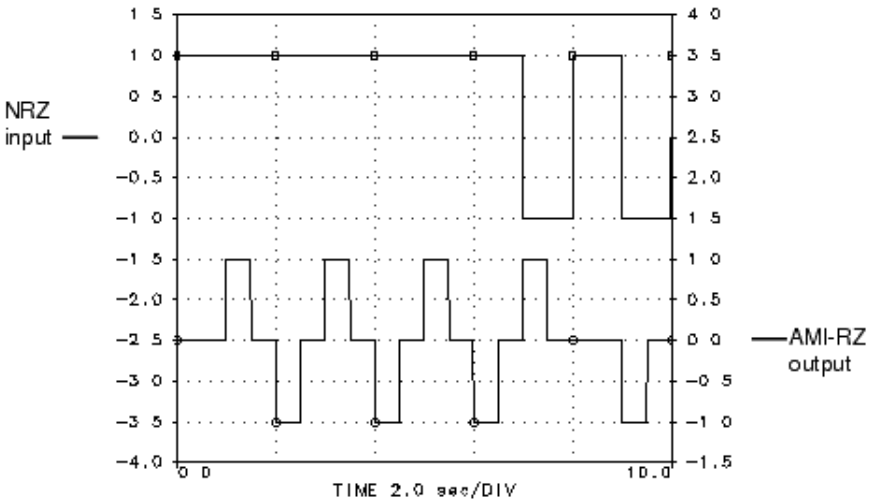
NRZ Input Versus CMI Output



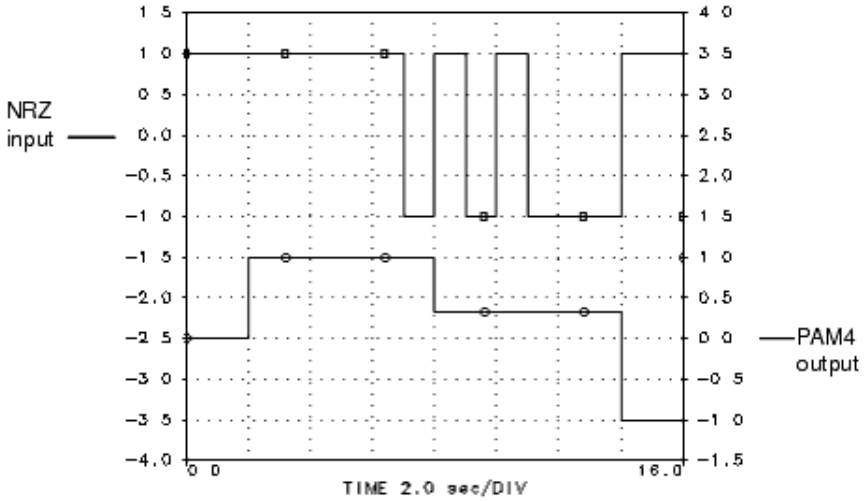
NRZ Input Versus AMI_NRZ Output



NRZ Input Versus AMI_RZ Output



NRZ Input Versus PAM4 Output



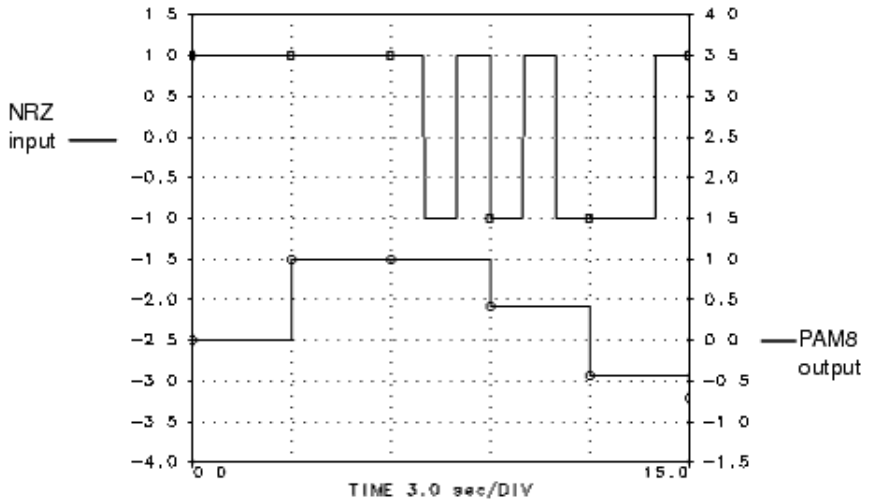
The SymbolConverter component in Agilent Ptolemy basically maps symbols in one format to symbols in another format. As an example, consider the case of NRZ input data and PAM4 output data as shown in NRZ Input Versus PAM4 Output above. The input symbols are taken two at a time, which can result in one of four states. The output value assumes one of these four states, as shown in *NRZ Input Verses PAM4 Output Example*.

NRZ Input Verses PAM4 Output Example

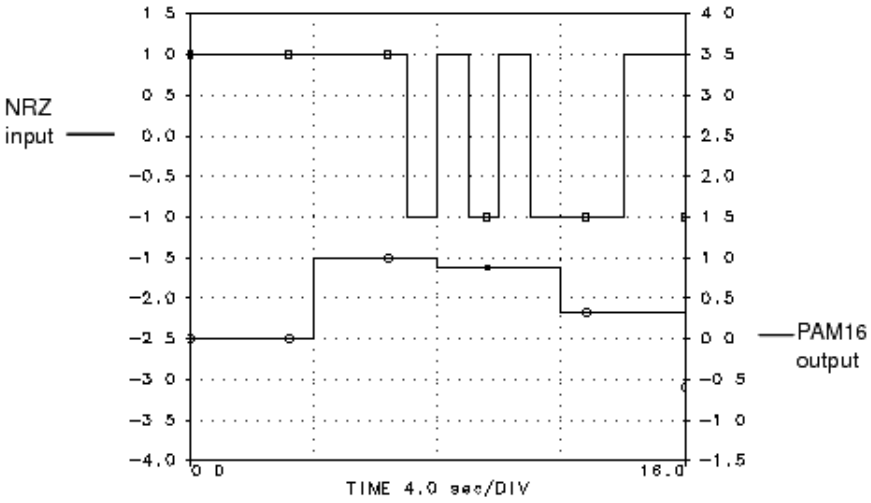
Two adjacent NRZ symbols	PAM4 value
0 0	-1.0
0 1	-0.33
1 0	0.33
1 1	1.0

Note that two consectutive NRZ symbols will produce two repeated PAM4 symbols. This is because the input symbols have 1 sample per symbol. The SymbolConverter converts two of these symbols in one PAM4 symbol. The value of the PAM4 symbol needs to be constant for the duration of the two input symbols. Since the converter does not change the time step, the PAM4 output symbol will have 2 samples per symbol. If you chose PAM8, the output PAM8 symbol would have 3 samples per symbol.

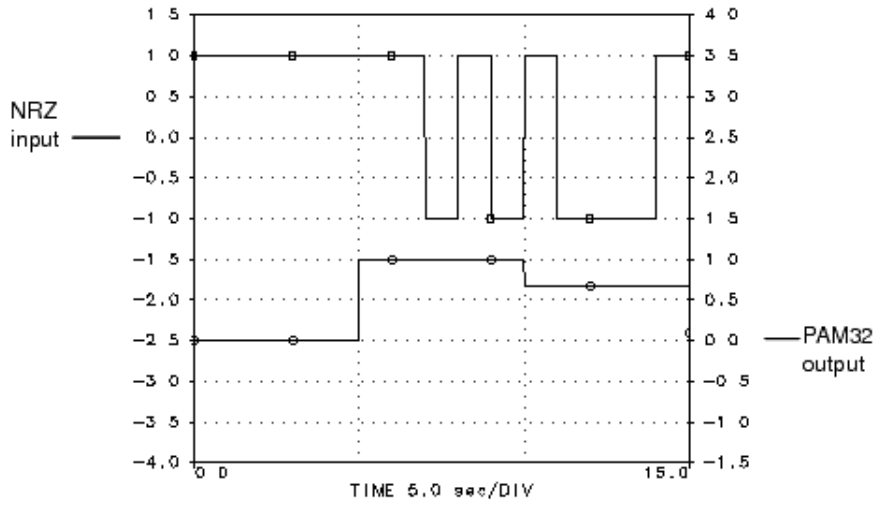
NRZ Input Versus PAM8 Output



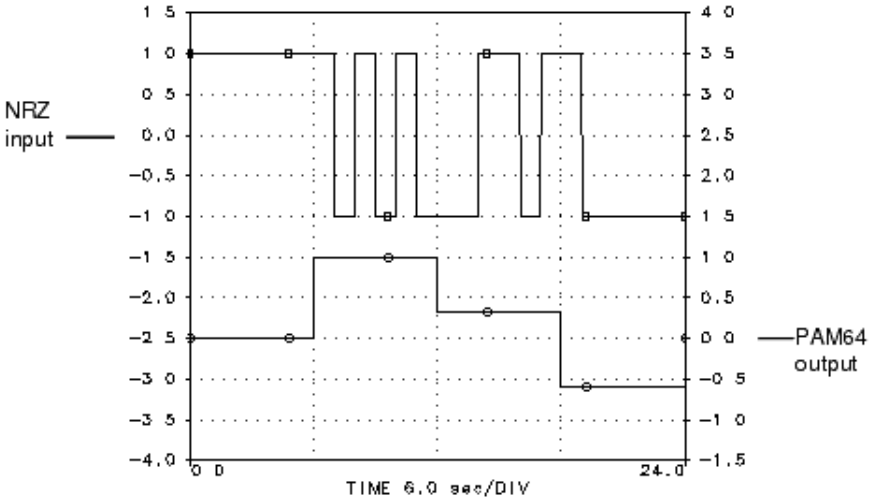
NRZ Input Versus PAM16 Output



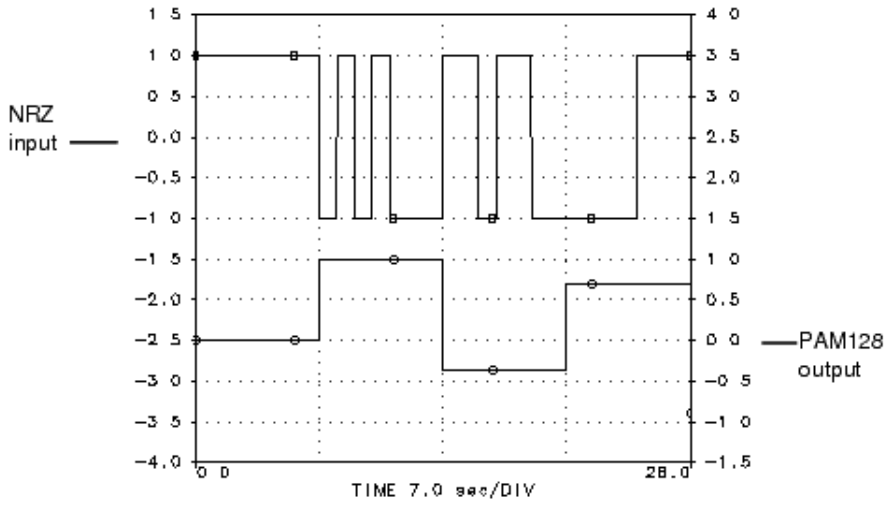
NRZ Input Versus PAM32 Output



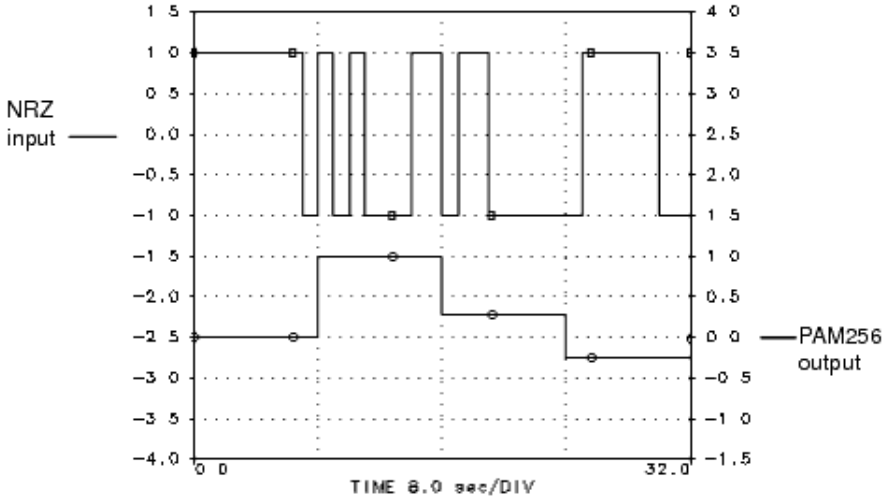
NRZ Input Versus PAM64 Output



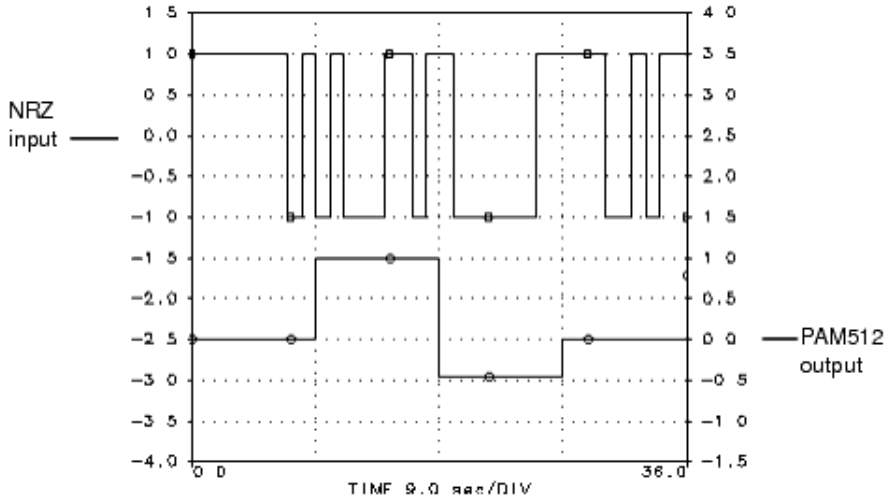
NRZ Input Versus PAM128 Output



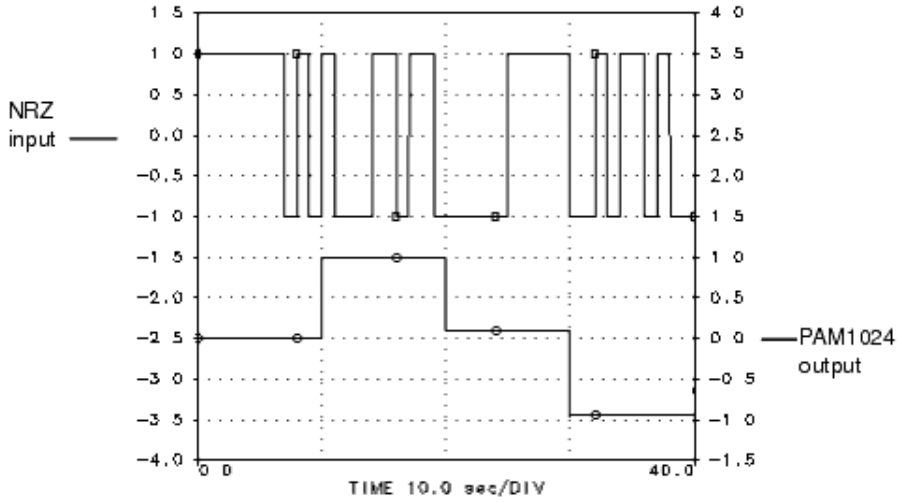
NRZ Input Versus PAM256 Output



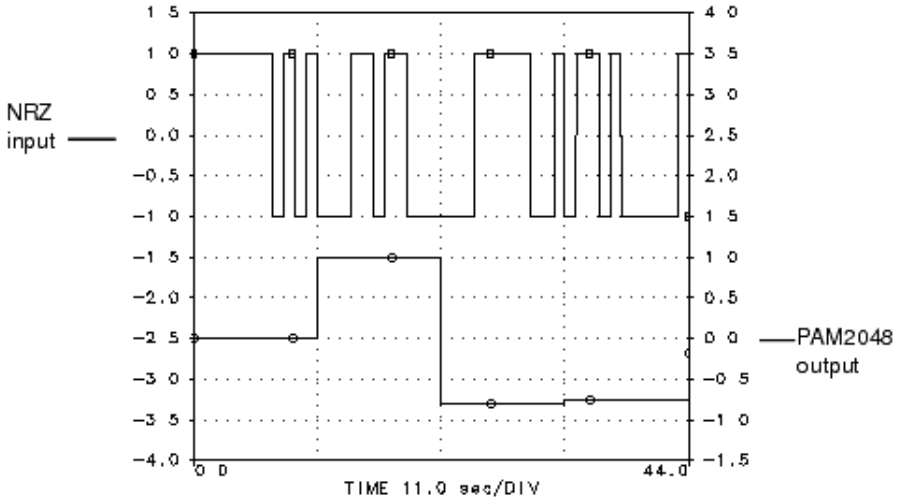
NRZ Input Versus PAM512 Output



NRZ Input Versus PAM1024 Output

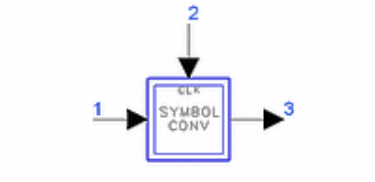


NRZ Input Versus PAM2048 Output



5. For information regarding timed data processing component signals, refer to *Timed Data Processing Components* (timed).

SymbolConverterClocked



Description: Symbol format converter with clock input
Library: Timed, Data Processing
Class: TSDF_SymbolConverterClocked
Derived From: baseOmniSysStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
CodeIn	input data format: nrzIn, rzIn, cmiIn, ami_nrzIn, ami_rzIn, pam4In, pam8In, pam16In, pam32In, pam64In, pam128In, pam256In, pam512In, pam1024In, pam2048In	nrzIn		enum	†
CodeOut	output data format: nrzOut, rzOut, cmiOut, ami_nrzOut, ami_rzOut, pam4Out, pam8Out, pam16Out, pam32Out, pam64Out, pam128Out, pam256Out, pam512Out, pam1024Out, pam2048Out	nrzOut		enum	†

† Either CodeIn or CodeOut must be NRZ format

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed
2	clock	clock signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
3	output	output signal	timed

Notes/Equations

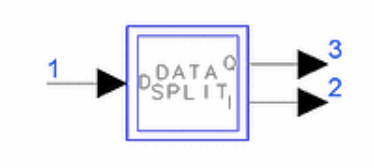
1. This component converts data from NRZ format to one of the other formats, or from one of the other formats to NRZ format. The two input signals are the input data stream $V_1(t)$ and a clock signal $V_2(t)$; the output signal is $V_3(t)$.

   If the CodeIn and CodeOut parameters are set to either nrz or pamn, then the clock period should equal the input symbol time. However, if either CodeIn or CodeOut is

set to one of the other data formats, then the input clock rate should equal twice the input symbol rate (that is, for a 1 µsec symbol rate, the clock period should be 0.5 µsec). And, for CodeIn or CodeOut, not nrz or pamn, then the first clock leading edge should align with the leading edge of the input symbols. The output $V_3(t)$ is set to 0V before the first positive edge of the input clock.
For representative examples on converting NRZ format to each possible output format, refer to SymbolConverter component documentation.

2. The output signal is causal; therefore the output is equal to the 0V initial condition until after the synchronization time.
3. For information regarding timed data processing component signals, refer to *Timed Data Processing Components* (timed).

SymbolSplitter



Description: Symbol splitter for alternate symbols
Library: Timed, Data Processing
Class: TSDF_SymbolSplitter
Derived From: baseOmniSysStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
SymbolTime	input data symbol time	0.001	sec	real	[TStep, ∞)†
Delay	data input time delay from t=0 for start of the data stream, Delay= -1 for auto synchronization	0	sec	real	{-1} or [0, ∞)††

† TStep is the simulation time step for the component input signal.
†† Refer to Note 2 for the special meaning of a -1 value.

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

Pin Outputs

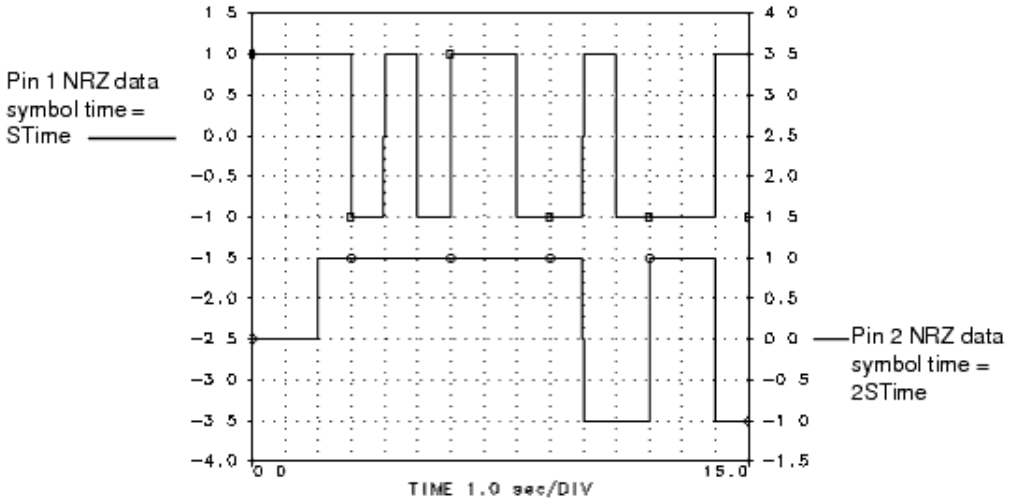
Pin	Name	Description	Signal Type
2	I_out	I output	timed
3	Q_out	Q output	timed

Notes/Equations

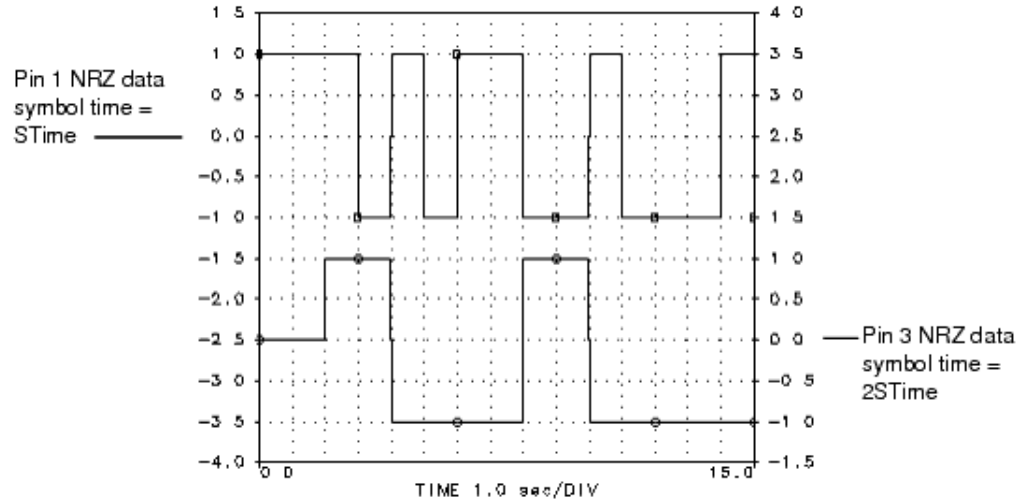
1. This component is used to convert an input bit stream into two bit streams (also refer to the SymbolSplitterClocked component).
   The input data signal is presumed to be in the NRZ binary format, with logic 0 level of −1V and logic 1 level at +1V. Output data streams are in the NRZ binary format with +1 and −1 levels and a symbol time of 2SymbolTime. Even input bits are sent to pin 2; odd input bits are sent to pin 3.
2. The data splitter performs an integrate and dump operation on the input data to determine the state of the input data at pin 1. The integration period is equal to SymbolTime. At the end of the integration period, a positive accumulation sets the

input data to +1, a negative accumulation sets the input data to −1. Delay is used to set the time instant at which the integration is started.
When Delay=−1, self-synchronization is performed. The program performs self-synchronization by determining the time at which either input's absolute voltage is greater than 0.5V. This is the time when the integrate and dump process begins. Previous to this instant, the outputs are set to 0.
Data Splitter Showing Pin 1 and Pin 2 Performance and Data Splitter Showing Pin 1 and Pin 3 Performance illustrate the data splitter input and output.

Data Splitter Showing Pin 1 and Pin 2 Performance

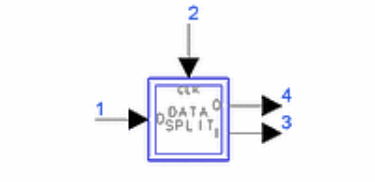


Data Splitter Showing Pin 1 and Pin 3 Performance



3. For information regarding timed data processing component signals, refer to *Timed Data Processing Components* (timed).

SymbolSplitterClocked



Description: Symbol splitter for alternate symbols with clock input
Library: Timed, Data Processing
Class: TSDF_SymbolSplitterClocked
Derived From: baseOmniSysStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed
2	clock	clock signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
3	I_out	I output	timed
4	Q_out	Q output	timed

Notes/Equations

1. The input data signal is presumed to be in the NRZ binary format with logic 0 level of −1V and logic 1 level of +1V.
2. This component is used to convert a serial NRZ data stream into two parallel NRZ data streams (also refer to *SymbolSplitter* (timed) component information). The even input data symbols are output on one output port; the odd input symbols are output on the other output port. At each positive edge of the input clock, the input signal is sampled and compared to a threshold voltage of 0V to determine its data state. At every alternate positive clock edge, the current and the previous data states of the input signal are output on the two output pins. Thus, the duration of the output bits is twice as long as that of the input bits.
   Mathematically the component can be described as:
   $V_1(t)$ is the input signal

   $V_2(t)$ is the input clock signal

   $V_3(t)$ and $V_4(t)$ are the output signals

   Let T_0, T_1, T_2 ... be the time instances when the positive edges of the input clock,

$V_2(t)$, occur (a positive edge occurs at the instant when the clock voltage changes from a negative to a non-negative value).

Let

$$A(k)=\begin{cases}1 & if\ V_1(T_K)\geq 0\\ -1 & if\ V_1(T_K)<0\end{cases}$$

Then

$V_3(t) = A_{2k}$ for $T_{2K+1} \leq t < T_{2K+3}$

$V_4(t) = A_{2k+1}$ for $T_{2K+1} \leq t < T_{2K+3}$

Note that the output signal is causal-therefore, the output is equal to the 0V initial condition until after synchronization time.

3. For information regarding timed data processing component signals, refer to *Timed Data Processing Components* (timed).

Timed Filters

The Timed Filter library contains Butterworth, Bessel, Chebyshev, Gaussian, Elliptic and Raised-Cosine response categories. A lowpass and a bandpass filter is available in each category.

The timed filter components consume and produce a timed signal. If a component receives another class of scalar signals, the received signal is automatically converted to a timed (baseband) type of signal. Auto conversion from the complex scalar to the timed class is not allowed. The user must explicitly use the CxToTimed signal converter for this transformation. These components do not accept any matrix class of signal.

Note
If the simulation time step is too large to enable proper characterization of the filter impulse response, the filter model will default to an all-pass model with only one non-zero FIR tap coefficient. A warning message will be displayed stating that this all-pass model will be used and will identify the maximum time step allowed to achieve the full impulse model.

For example: a LPF with passband 1 MHz but a signal time step of 1 msec will cause the filter model to revert to an impulse. This is acceptable since the signal information bandwidth is 1 kHz and the LPF will appear as an all-pass to this signal. Thus the impulse model is justified. As this time step decreases below 1 msec, the LPF characteristic will begin to affect the signal and the actual filter characteristic will be engaged.

The general rule is that the time step should be less than 0.25/(LPF_3DB_BW) for N=5. This gives a simulation bandwidth of 2*LPF_3DB_BW. This upper limit typically needs to be lower as the filter order (N) approaches 1 and can be larger as N increases above 5. The filter StopBW/StopAtten specification is always translated to the N required before the filter impulse model is calculated.

If the simulation time step is small enough to characterize the filter impulse response, the following impulse modeling steps are used.

All filters are modeled as a finite impulse response (FIR) type of filter. Each filter has an impulse response h(t) calculated from an exact time domain impulse model-except the raised-cosine model with pulse equalization, which is modeled in the frequency domain and is converted into an equivalent time domain model. For the raised-cosine filter models, the user may specify a (time) Delay value resulting in a filter impulse model that is twice as long as the delay specified; for all other filters, the user may specify an impulse model time duration (ImpTime).

All filters have user-specified input (RIn) and output resistance (ROut). The input resistance is for a shunt resistor at the component input pin; the output resistance is for a series resistor at the component output pin.

Note
Prior to the 2003A release, bandpass filter components did not produce the correct output voltage level when their input resistance was not 50 Ohms. This problem is fixed in the 2003A release; a change (compared to previous releases) in the output signal voltage level may be observed if RIn was not set to 50 Ohms.

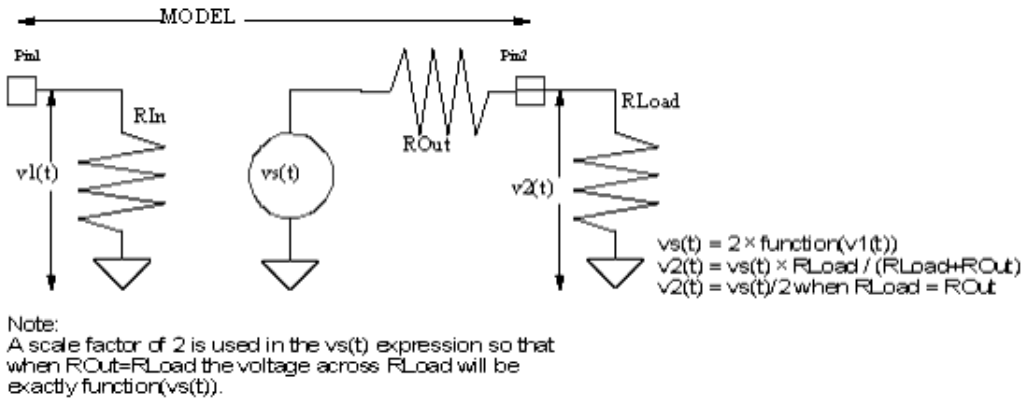
All filters have user-specified power loss, Loss, (in dB) with respect to matched input and output resistance loading conditions.

Given an input time domain signal at the input pin v1(t) and the filter impulse model h(t), the output signal source vs(t) is obtained from the convolution of the input signal and the impulse model.

When the input is a timed baseband signal, the impulse model is real and the output signal is also a timed baseband signal. When the input is a timed complex envelope signal, the impulse model is complex and the output signal is a timed complex envelope signal with carrier frequency equal to that for input.

The output signal source vs(t) has a value equal to twice the output pin signal v2(t) when the output is connected to a matched resistor load. The circuit model shown in Circuit Model demonstrates this relationship.

Circuit Model



The output pin signal v2(t) at the output series resistance is dependent on the value of the load resistance connected to it. When the load resistor RLoad is equal to the model output resistor ROut the value of v2(t) is equal to vs(t)/2; otherwise, based on the voltage divider action, v2(t) is:

$$v2(t) = loss \times vs(t) \times RLoad/(RLoad + ROut)$$

where loss = $10^{(-abs(Loss)/20)}$ and the input and output resistor values must be greater than 0 ohm.

Only Loss contributes additive thermal noise power ($kT_e B$) to the output signal when the specified resistance temperature (RTemp) is greater than absolute zero (−273.15 °C) and Loss (in dB) is greater than 0, where:

k = Boltzmann's constant

T_e = equivalent noise temperature in Kelvin = (1 - ploss) (RTemp +273.15)

B = simulation frequency bandwidth:

1/2/tstep if signal is a timed baseband signal;

1/tstep if signal is a timed complex envelope signal

ploss = $10^{(-abs(Loss)/10)}$

When RTemp > −273.15, the noise contributed by the Loss is an independent noise process. This noise is dependent on the value of DefaultSeed in the DF (data flow) controller. When DefaultSeed=0, then the noise generated for each simulation is different. When DefaultSeed>0, then the noise generated for each simulation, though random, has the same initial seed starting condition and thus results in reproducible noise for each simulation.

All filter responses are based on lowpass prototypes with −3 dB point at 1 Hz. From these prototypes, transformations are used to obtain all lowpass and bandpass responses.

Filter component characteristics are described in the following paragraphs. Although lowpass characteristics are illustrated, the comments also apply to bandpass filters.

Note
Information regarding time domain signal differences between ADS Ptolemy simulations and Circuit Envelope and Transient simulations is given in the *Timed Synchronous Dataflow* (ptolemy) section in the *ADS Ptolemy Simulation* (ptolemy) documentation.

Butterworth Filter Characteristics

The Butterworth response filter family, also known as maximally-flat magnitude response filters, are typically used when a compromise between good selectivity and good group delay flatness is desired. Butterworth Magnitude and Group Delay Response shows Butterworth lowpass filter magnitude and group delay response for filter orders of 1-5, 10, and 15-all with corner attenuation of 3 dB.

Bessel Filter Characteristics

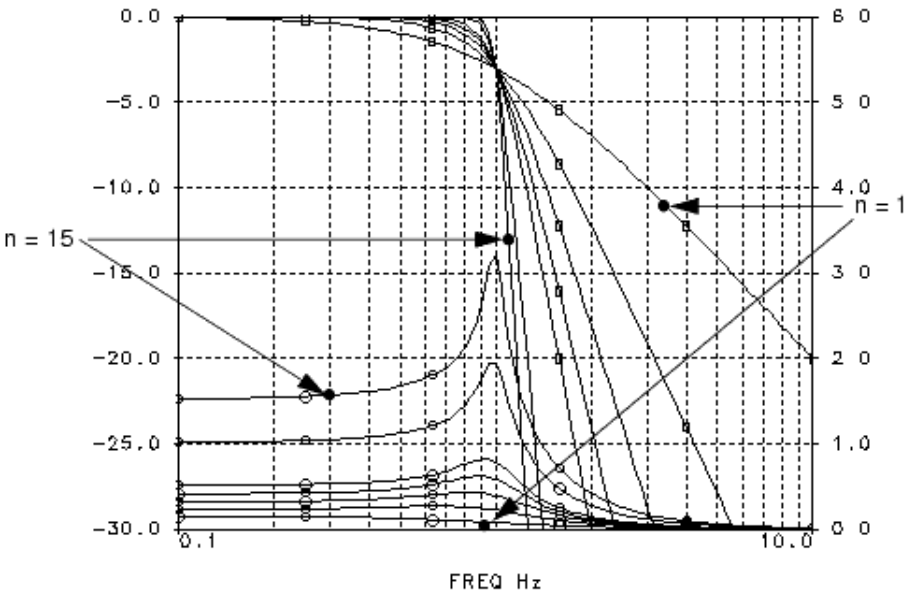
Bessel filters are derived from a maximally-flat delay criterion. As can be seen by comparing Butterworth Magnitude and Group Delay Response and Bessel Magnitude and Group Delay Response, the Bessel has better group delay flatness than the Butterworth, but at the cost of a less sharp corner. These filters are often used where group delay flatness is important, as in FM video transmission.

Bessel Magnitude and Group Delay Response shows Bessel lowpass filter magnitude and group delay response for filter orders of 1 through 5 and 8-all with a corner attenuation of 3 dB.

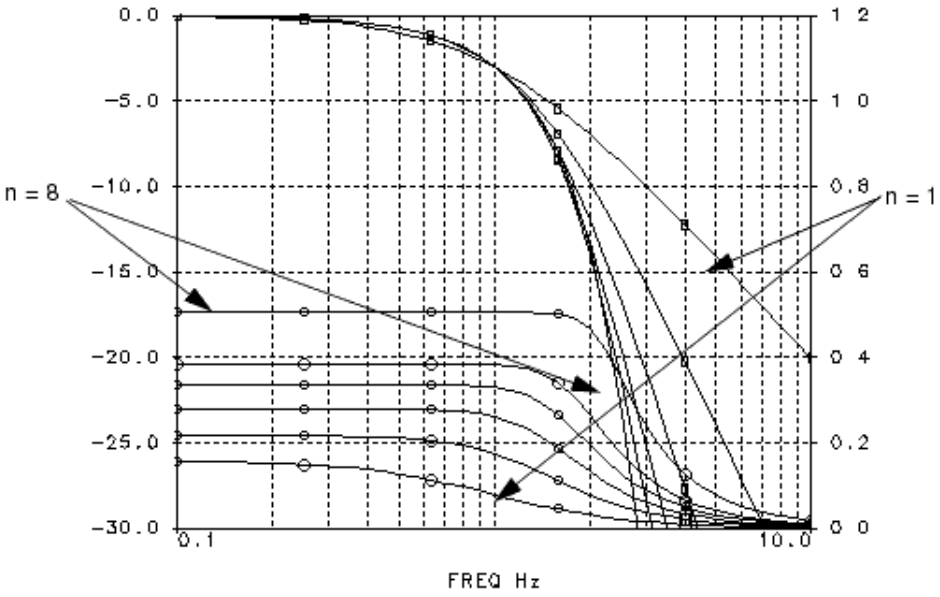
Chebyshev Filter Characteristics

Filters in the Chebyshev category are used to provide sharp corner frequency domain characteristics. Chebyshev In-Band and Out-of-Band Magnitude Response shows the in-band and out-of-band magnitude response of Chebyshev lowpass filters with 1 dB ripple, for filter orders 1through 5, 8 and 15. Chebyshev Group Delay Response shows the corresponding group delay response. The higher the filter order and ripple, the sharper the corner will be; however, sharper corner leads to worse phase linearity.

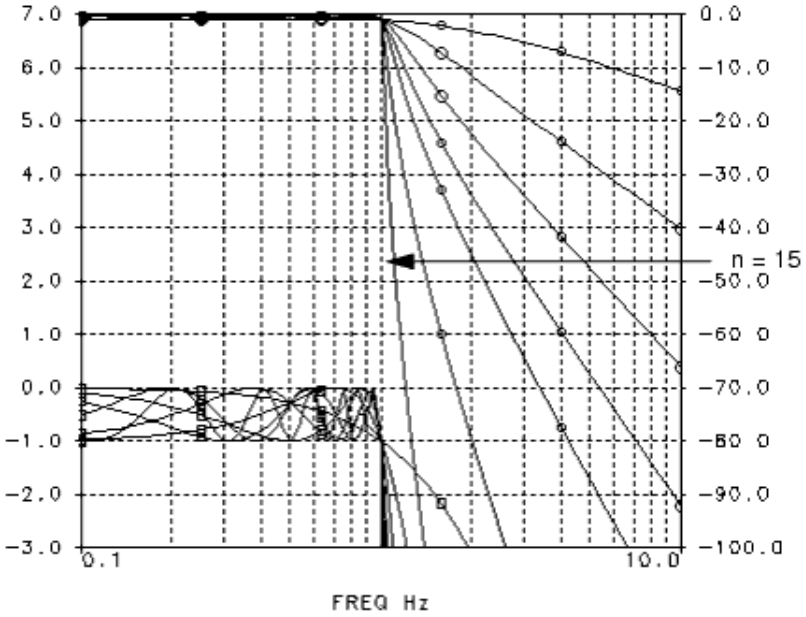
Butterworth Magnitude and Group Delay Response



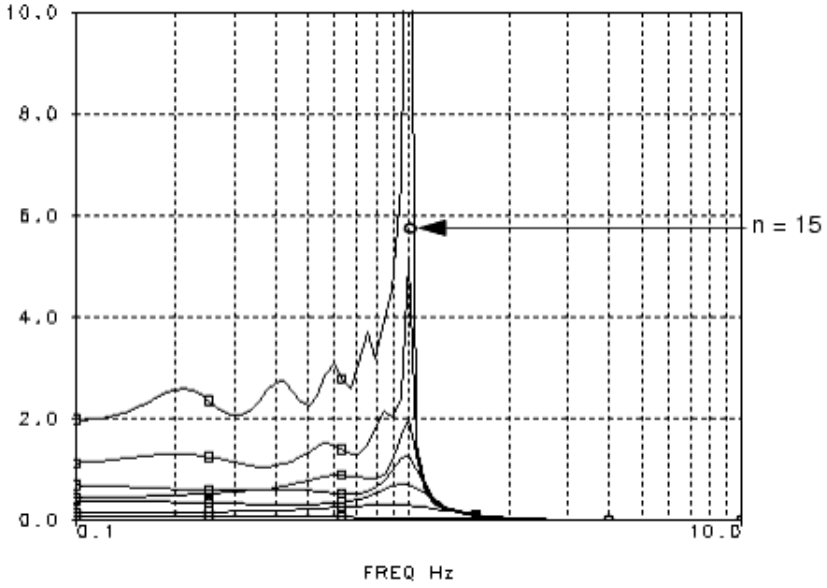
Bessel Magnitude and Group Delay Response



Chebyshev In-Band and Out-of-Band Magnitude Response



Chebyshev Group Delay Response


10.0
8.0
6.0
4.0
2.0
0.0
0.1
10.0
n = 15
FREQ Hz

Elliptic Filter Characteristics

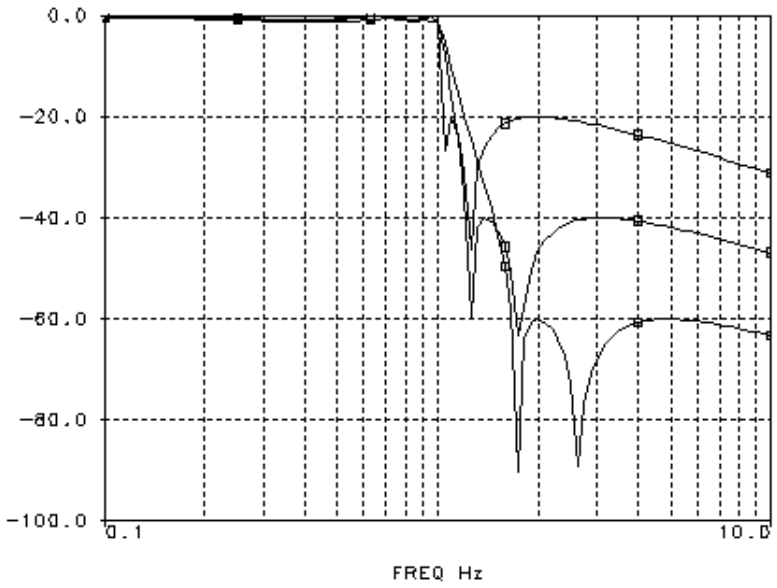
Elliptic filters have even sharper corner response characteristics than Chebyshev filters because of the inclusion of out-of-band zeros. Elliptic Filters Sharp Magnitude Corner Response illustrates the elliptic response, showing fifth order filters with 1 dB ripple, and rejection of 20, 40, and 60 dB.

The elliptic filters (also referred to as Cauer filters) have extremely nonlinear phase near the corner frequency, as can be seen from Elliptic Filter with Extremely Nonlinear Phase Near Corner Frequency. This shows group delay for the same filters as in Elliptic Filters Sharp Magnitude Corner Response. However, note that elliptic filters have better in-band group delay flatness than Chebyshev filters, partly because their sharper corner allows a filter of lower order to be used.

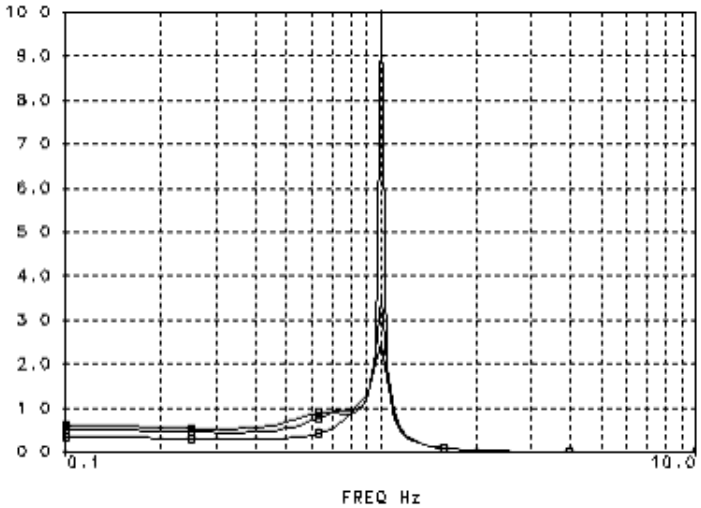
Gaussian Filter Characteristics

Gaussian filters have the special pulse filtering property, that is, they provide the fastest pulse rise time with no overshoot or ringing in the time domain. The ideal Gaussian filter characteristic is used. The filter is specified by the corner attenuation frequency, corner attenuation, and additional parameters. Gaussian Filter Magnitude Response shows the ideal Gaussian lowpass filter magnitude response with a corner attenuation of 3 dB.

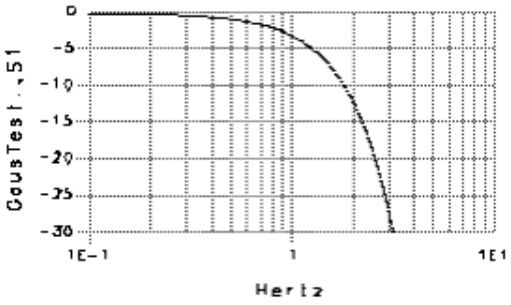
Elliptic Filters Sharp Magnitude Corner Response



Elliptic Filter with ExtremelyNonlinear Phase Near Corner Frequency



Gaussian Filter Magnitude Response



Raised-Cosine Filter Characteristics

Raised-cosine filters are used for shaping pulses for transmission through digital channels to prevent intersymbol interference (ISI) [3,4]. Rather than filter order, ripple, and so on, these filter elements are specified using the parameters CornerFrequency, ExcessBW, Exponent E=1 if SquareRoot=NO or E=0.5 if SquareRoot=YES, and Type, defined as follows.

The excess bandwidth beyond the corner frequency is traded off for a more easily implemented filter.
Let F = specified corner frequency of the filter, then the filter zero transmission cutoff frequency, fc, is
fc = F (1 + A)
and A = excess bandwidth.
Typically, for data transmission with symbol rate R, the value of F is set equal to R/2.
The transfer function of this filter is determined as follows:
Type = Impulse model

$$H(j\omega) = (H_{RC}(j\omega))^{E} e^{-j2\pi fDelay}$$

Type = Model with pulse equalization

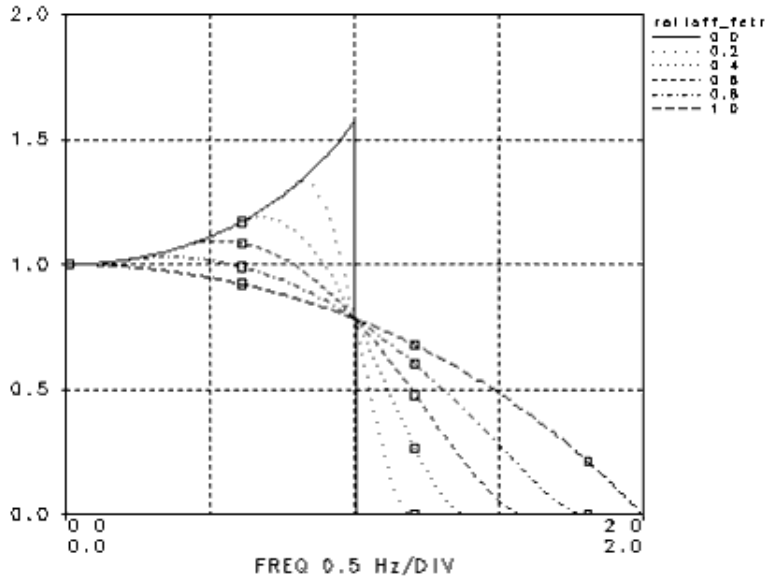
$$H(j\omega) = \frac{(\omega/(4F))}{\sin(\omega/(4F))}(H_{RC}(j\omega))^{E} e^{-j2\pi fDelay}$$

where $H_{RC}(j\omega)$ is the transfer function of a raised-cosine filter:

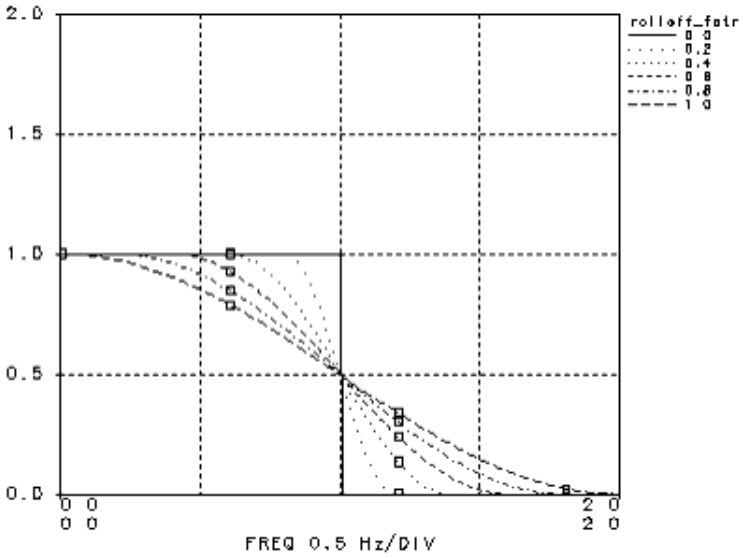
$$H_{RC}(j\omega) = \begin{cases} 1 & \text{for } 0 \le \omega \le 2\pi F(1-A) \\ \left(\cos\left(\frac{\omega}{8FA} - \frac{\pi(1-A)}{4A}\right)\right)^2 & \text{for } 2\pi F(1-A) \le \omega \le 2\pi F(1+A) \\ 0 & \text{for } \omega \ge 2\pi F(1+A) \end{cases}$$

Ideal Raised-Cosine Pulse Response with Equalized Sinc-Shaped Spectrum shows the frequency response of lowpass raised-cosine filters with E=1, Type=Model with pulse equalization and a parametrized rolloff factor A from 0 to 1 in steps of 0.2. Ideal Raised-Cosine Impulse Response is the same except Type=Impulse model.

Ideal Raised-Cosine Pulse Response with Equalized Sinc-Shaped Spectrum



Ideal Raised-Cosine Impulse Response



Frequency Transformations

In accordance with standard practice, the lowpass and bandpass filter models result from performing the appropriate frequency transformation on the lowpass prototype. For details, refer to [1]. In all cases, the program filter component uses the standard wideband transformation. For reference, the transformations used are
Lowpass to lowpass:

$$\Omega = \frac{\omega}{\omega_c}$$

Bandpass to lowpass:

$$\Omega = \frac{1}{\gamma}(|\omega - \omega_o|)$$

where
Ω = lowpass frequency
ωc = radian frequency at −3dB attenuation
ω = radian frequency
γ = relative bandwidth of −3dB attenuation frequencies
ωo = center frequency

Note
The bandpass-to-lowpass transformation results in symmetrical amplitude response above and below the center frequency and closely approximates the usual analog filter bandpass-to-lowpass transformation

$$\Omega = \frac{1}{\gamma}\left(\frac{\omega}{\omega_o} - \frac{\omega_o}{\omega}\right)$$

when the relative bandwidth is small.

Data Flow Analysis

For data flow analysis, the impulse response of the filter is derived prior to simulation. An FIR filter with tap coefficients equal to the impulse response samples is created internally. The following paragraphs describe how the filters are modeled.

First, the impulse response of the filter is computed. The user can control the number of samples of the impulse response through *ImpTime* or *Delay* parameters. The number of samples of the impulse response (number of filter taps) is given by *ImpTime* / *TStep* or *Delay* / *TStep*, where *TStep* is the simulation time step of the input signal.

Then, the impulse response obtained from the first step is windowed. The default is a rectangular window, but the user can choose between Bartlett, Hanning, Hamming, and Flat Top windows (by applying a window, the stop band rejection of the filter can be increased, but at the expense of increasing the transition bandwidth). The following equations define the windowing operations.

Let the original impulse response be denoted as $h(kT_S)$, $0 \le k <$ N where T_S is the sampling interval and N is the number of points in the impulse response. The resulting windowed impulse, $h_w(kT_S)$ is given by the equation

$$h_w(kT_S) = w(kT_S)h(kT_S), \quad 0 \le k < N$$

where $w(kT_S)$ denotes the windowing function specified by WindowType.

Rectangular window:

$$w(kT_s) = 1.0 \quad 0 \le k < N$$

Bartlett (triangular) window:

$$w(kT_s) = \begin{cases} \frac{2k}{N} & 0 \le k < \frac{N}{2} \\ 2 - \frac{2k}{N} & \frac{N}{2} \le k < N \end{cases}$$

Hanning window:

$$w(kT_s) = 0.5 - 0.5\cos\left(\frac{2\pi k}{N}\right) \quad 0 \le k < N$$

Hamming window:

$$w(kT_s) = 0.54 - 0.46\cos\left(\frac{2\pi k}{N}\right) \quad 0 \le k < N$$

Flat Top window:

$$w(kT_s) = \begin{cases} \frac{1}{4.64}\left(1 - 1.93\cos\left(\frac{2\pi k}{N}\right) + 1.29\cos\left(\frac{4\pi k}{N}\right)\right. \\ \left. - 0.388\cos\left(\frac{6\pi k}{N}\right) + 0.0322\cos\left(\frac{8\pi k}{N}\right)\right) & 0 \le k \le N \end{cases}$$

In the third step, the impulse response obtained from the second step may have to be further modified depending upon the input signal. Let the signals at pins 1 (the input) and 2 (the output) be of the form

$$V_k(t) = Re\left\{v_k(t)e^{j2\pi f_{ck}t}\right\}, \quad v_k(t) = v_{Ik} + jv_{Qk}, \quad k = 1, 2$$

Lowpass Filter Case 1

If the input signal is in the baseband representation ($f_c 1 = 0$) the output is obtained by directly convolving the impulse response h _w _ (t) with the input signal:

$$V_2(t) = h_w(t) \otimes V_1(t)$$

and $f_c 2 = 0$.

Lowpass Filter Case 2

If the input signal is an RF (complex envelope) bandpass signal ($f_c 1 > 0$) the complex envelope $\tilde{h_w}(t)$ of the impulse response is first calculated:

$$\tilde{h_w}(t) = (h_w(t) + j\hat{h}_w(t))e^{-j2\pi f_{c1}t}$$

where $\hat{h}_w(t)$ is the Hilbert transform of h _w _ (t).

The output of the filter is then calculated:

$$v_2(t) = \tilde{h_w}(t) \otimes v_1(t)$$

and $f_c 2 = f_c 1$.

The above algorithm assumes that the condition ($f_c 1 > 0.5$ / TStep) is satisfied, where $f_c 1$ is the input signal characterization frequency and TStep is the simulation time step. If this condition is not satisfied, the output signal may be in error (depending on how severely the condition is violated). If the condition is several violated (for example, $f_c 1 =$ 30 MHz and 0.5/TStep = 150 MHz) it is recommended that the input signal be converted to a baseband representation by using the FcChange component.

Bandpass Filter Case 1

If the input signal is in the baseband representation (f_{c} 1 = 0) the assumption is made that the input signal is a bandpass signal with no significant energy at 0 Hz. The inphase and quadrature phase signals are then extracted by multiplying the input baseband signal by cos(2π Fc t) and (-1)sin(2π Fc t) and filtering the resulting signals with a lowpass filter of bandwidth Bandwidth (the default value of Bandwidth is FCenter). Note that a delay is introduced in this case because of the filtering. With the signal now defined as an RF (complex envelope) bandpass signal (new f_{c} 1 > 0) the complex filtering is performed in same manner as described next.

Bandpass Filter Case 2

If the input signal is an RF (complex envelope) bandpass signal (f_{c} 1 > 0) the complex signal filtering is performed by filtering the I and Q envelopes with equivalent lowpass filters. The input signal is first represented at the bandpass filter center frequency and the I and Q envelopes are filtered by the equivalent lowpass filter for the bandpass filter model at its center frequency as was done in the Bandpass Filter Case 1. The resultant filtered I and Q waveforms are combined to compose the resultant output complex envelope at the bandpass filter center frequency. This output signal is then represented at the input signal characterization frequency, f_{c} 1.

The input signal V_{1} (t) at pin 1 is represented by its inphase and quadrature components about its carrier frequency:

$$V_1(t) = Re\left\{v_1(t)e^{j2\pi f_{c1}t}\right\},\ v_1(t) = v_{I1}(t) + jv_{Q1}(t)$$

This input signal has an equivalent representation at bandpass filter center frequency

$$V_1(t) = Re\left\{v_1(t)e^{j2\pi f_{c1}t}e^{(-j)2\pi f_{bp}t}e^{j2\pi f_{bp}t}\right\} = Re\left\{v_{1P}(t)e^{j2\pi f_{bp}t}\right\}$$

The equivalent lowpass filtering is done on real and imaginary parts of $v_{1 \sim \sim \_P \_}$ (t) by directly convolving the impulse response $h_{\_w \_}$ (t) with the $v \sim 1 \sim_{\_P \_}$ (t) signal.

$$V_2(t) = Re\left\{(h_w(t) \otimes v_{1P}(t))e^{j2\pi f_{bp}t}e^{(-j)2\pi f_{c1}t}e^{j2\pi f_{c1}t}\right\}$$

This filtered complex signal is represented again at the input signal carrier frequency

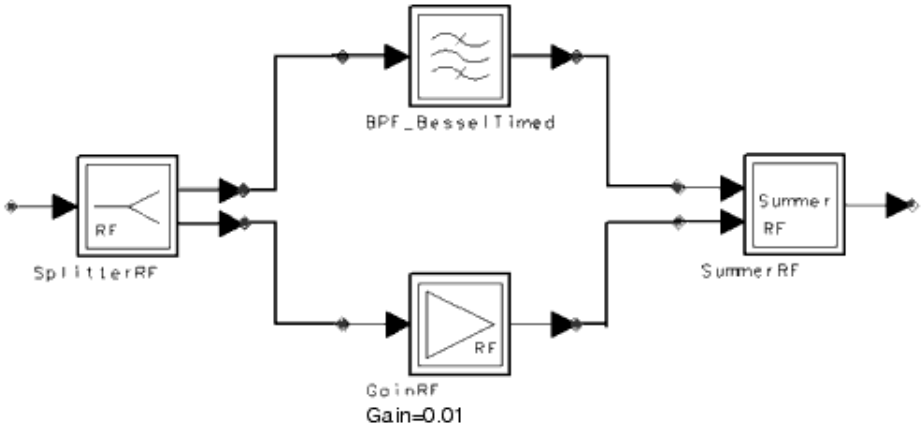
$$V_2(t) = Re\left\{v_2(t)e^{j2\pi f_{c1}t}\right\},\ v_2(t) = v_{I2}(t) + jv_{Q2}(t)$$

Note
The effect of the filtering on the input RF (complex envelope) bandpass signal results in delay and filtering of the I and Q envelopes, but there is no associated phase shift of the input RF carrier itself.

Additional Filter Characteristics

Timed filter components can be supplemented with additional characteristics such as maximum rejection specification or multiple passbands by using these filters in hierarchical designs. For example, to achieve a maximum rejection specification, arrange a bandpass filter in a schematic design with a parallel path for a GainRF component with SplitterRF and SummerRF components as in the following example (GainRF Component with SplitterRF and SummerRF Components). Here, the maximum rejection can be set to 0.01 (−40 dB) by the GainRF component. Similarly, other timed filter characteristics can be supplemented with other designs.

GainRF Component with SplitterRF and SummerRF Components



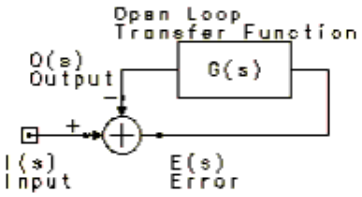
Second-Order Control Loop Filters

Second-order control loop filters typically have a pole and zero and gain with values set by second-order control loop performance requirements. This section describes loop filter parameter values derived from control loop requirements.

Basic Linear Control Loop

Though control loops can have general topologies, we can obtain useful design guidelines by studying the simple control loop shown in Simple Linear Control Loop.

Simple Linear Control Loop



You can compare the performance of such a control loop with the performance of other more complex and nonlinear control loops. Simple Linear Control Loop shows an input signal I, and output signal O, an error signal, E, and an open loop transfer function, G(s).

The closed loop transfer function is derived as follows:

$$H(s) = \frac{O(s)}{I(s)} = \frac{G(s)}{1 + G(s)}$$

The characteristic of the open loop transfer function, G(s), provides a classification that conveniently categorizes various control loop implementations. For our purposes, a second-order control loop is of interest. The loop *order* specifies the total number of poles in G(s). Additionally, the *type* of a loop is specified by the number of poles at the origin in G(s).

The table below shows the basic control loop transfer functions. The parameter *K* in the open loop transfer function G(s) is called the loop gain.

Loop Order	Loop Type	Open Loop G(s)	Closed Loop H(s)
1	0	$\frac{K}{s+a}$	$\frac{K}{s+a+K}$
1	1	$\frac{K}{s}$	$\frac{K}{s+K}$
2	0	$\frac{Kbc(s+a)}{a(s+b)(s+c)}$	$\frac{Kbc(s+a)/a}{s^2+s(b+c+Kbc/a)+bc(1+K)}$
2	1	$\frac{Kb(s+a)}{a(s+b)s}$	$\frac{Kb(s+a)/a}{s^2+s(b+Kb/a)+Kb}$
2	2	$\frac{K(s+a)}{s^2}$	$\frac{K(s+a)}{s^2+sK+Ka}$

Control Loop Equations

Second-order linear control loops are often used as approximations for actual RF control loops. This section presents the equations defining these control loops in terms of their frequency-domain and time-domain performance parameters.

The second-order, type 1 closed loop response, in terms of loop natural frequency (ωn), loop damping factor (ζ), and closed loop zero (δ), can be expressed as:

$$H(s) = \frac{O(s)}{I(s)} = \frac{(s/\delta + 1)\omega_n^2}{s^2 + 2\varsigma\omega_n s + \omega_n^2}$$

where the following associations are made with the related expression from *Basic Control Loop Transfer Functions*.

$$\omega_n = \sqrt{Kb}$$

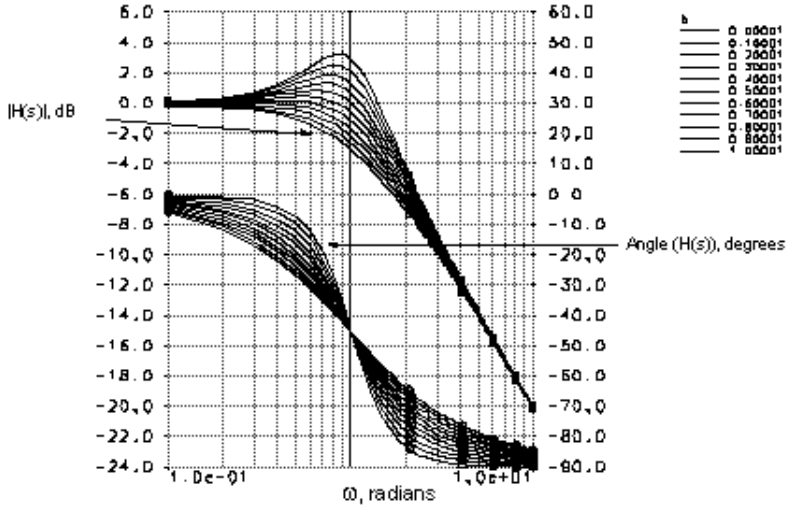
$$\varsigma = \frac{b + Kb/a}{2\sqrt{Kb}}$$

$$\delta = a$$

Second-Order, Type 1 Linear Frequency Domain Analysis

For the second-order, type 1 control loop, Closed Loop Frequency Response, Second-Order Type 1 Loop shows the closed loop frequency domain magnitude (in dB) and phase response versus frequency normalized to the loop natural frequency, as a function of the loop filter pole (b) normalized to the loop natural frequency. In this figure, the closed loop zero (δ) is set equal to the loop natural frequency.

Closed Loop Frequency Response, Second-Order Type 1 Loop



$$H(s) = \frac{kb(s+a)/a}{s^2 + s(b + kb/a) + kb}$$

$$a/\omega_n = 1; b/\omega_n = 0, 0.1, 0.2, \ldots 1.0$$

Additional frequency domain properties for the second-order type 1 control loop are as follows (from *Linear Control Systems* by J. L. Melsea and D. G. Schultz, McGraw-Hill, Inc., New York, 1969).

- fn, loop natural frequency, Hz
  fn = ωn/2/PI
- FP, closed loop frequency for maximum gain
  $FP = \sqrt{-1 + \sqrt{1 + (A + 1 - 2\varsigma^2)/\delta^{*2}}}\,\delta/2/\pi$

$$A = 1 + 1/\delta^{*2} - 2\varsigma^2$$

$$\delta^* = \delta/\omega_n$$

- AP, maximum closed loop gain at frequency FP
  $AP = \omega_n^2\sqrt{1 + (2\pi FP/\delta)^2}/D$

$$D = \sqrt{(\omega_n^2 - (2\pi FP)^2)^2 + (4\varsigma\omega_n\pi FP)^2}$$

- BW, closed loop response -3 dB frequency, Hz
  $BW = f_n\sqrt{A + \sqrt{A^2 + 1}}$

$$A = 1 + 1/\delta^{*2} - 2\zeta^2$$

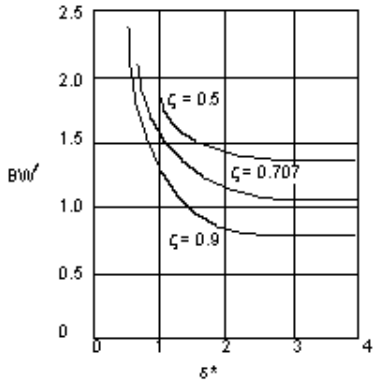
$$\delta^* = \delta/\omega_n$$

- BWn, closed loop response noise bandwidth, Hz

$$BWn = \frac{\omega_n(\omega_n^2/\delta + \delta)}{8\delta\zeta}$$

A plot of the second-order, type 1, BW'=BW/fn versus ζ and δ* is shown in BW=BW/fn versus z and d*, Second-Order Type 1 Loop.

BW'=BW/fn versus ζ and δ*, Second-Order Type 1 Loop



Second-Order, Type 1 Linear Time Domain Analysis

The second-order, type 1 control loop, has time properties as follows.

For step input:
I(t) is a step input
O(t) is the output closed loop step response
Let

$$\delta^* = \delta/\omega_n$$

$$t^* = \omega_n t; \quad \text{t=time}$$

$$s^* = s/\omega_n$$

Then

$$O(s^*) = 1/\delta^*(s^* + \delta^*)/s^*/(s^{*2} + 2\zeta s^* + 1)$$

and

$$O(t^*) = 1 + \frac{\sqrt{1 + \delta^*(\delta^* - 2\zeta)}}{\delta^*\sqrt{1-\zeta^2}}\exp(-\zeta t^*)\sin(\sqrt{1-\zeta^2}t^* + \varphi)$$

resulting in
t*ms normalized time for maximum step overshoot

$$t^*\text{ms} = \frac{\pi + arc\sin(\sqrt{1-\zeta^2}) - \varphi}{\sqrt{1-\zeta^2}}$$

POs percent step overshoot at t*ms

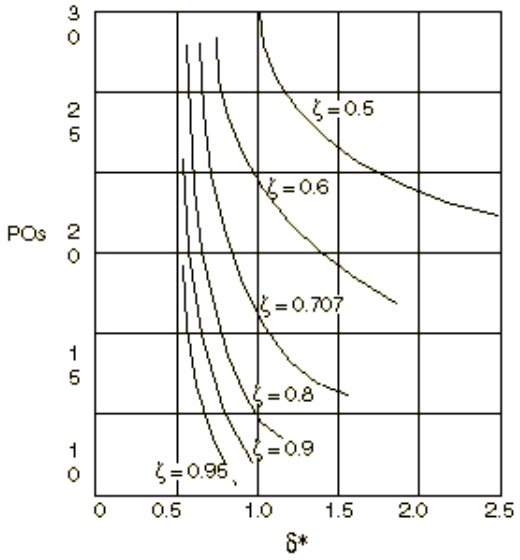
$$POs = 100\sqrt{1 + 1/\delta^{*2} - 2\zeta/\delta^*}\exp(-\zeta\phi/\sqrt{1-\zeta^2})$$

where

$$\varphi = arctan(\sqrt{1-\zeta^2}/(\delta^* - \zeta)) - arctan(\sqrt{1-\zeta^2}/(-\zeta))$$

$$\phi = \pi - arctan(\sqrt{1-\zeta^2}/(\delta^* - \zeta))$$

A plot of POs versus ζ and δ* is shown in Percent Overshoot (POs), Versus z and d*, Second-Order Type 1 Loop for the second-order type 1 loop.



Percent Overshoot (POs), Versus ζ and δ*, Second-Order Type 1 Loop

Loop Filters

The open loop transfer function G(s), can be represented in terms of a detector gain factor, KD, a voltage gain factor, KV, a loop filter, F(s), and possibly an integrator. The loop filter provides the poles and zeros of G(s).

Loop filters may be categorized on the basis of the filter order and number of filter zeros and integrators. The integrators are, in fact, poles and are included in the count of poles for the filter order. The loop filter gain is listed as K_F .

The loop filter is typically a circuit network composed of resistors, capacitors, and (possibly) op amps. The op amp model typically includes a single dominant pole and time delay:

$$V_s = (V^+ - V^-)M\frac{e^{-j2\pi fT}}{1 + j2\pi f/F}$$

where

V_s = op amp output voltage

V^+ = noninverting input voltage

V^- = inverting input voltage

M = magnitude of the voltage gain at dc
T = op amp time delay
F = frequency at which the gain magnitude is down by 3 dB
f = simulation frequency
Additionally, the op amp model includes input, output, and leakage resistances.

Though all of these op amp characteristics may be included within the control loop simulation, the following discussion on loop filters considers the op amp to have ideal infinite gain for simplification of the technical presentation.

The table below lists various loop filter designs. Parameters *a* and *b* are filter poles and zeros; parameters τ1 and τ2 are time constants.

***Filter Design, Order, Zeros,**	**Transfer Function with Poles and Zeros**	**Transfer Function with Time Constants**
F211(s)	$\frac{K_F b(s+a)}{as(s+b)}$	$\frac{K_F(s\tau_1+1)}{s(s\tau_2+1)}$
F110(s)	$\frac{K_F b(s+a)}{a(s+b)}$	$\frac{K_F(s\tau_1+1)}{s\tau_2+1}$

Basic second-order control loop equations are shown next.
Basic Second-Order Control Loop Equations (Order 2, Type 1)
I(s) = input
O(s) = output
E(s) = loop error = I(s) - O(s)
HOL(s) = open loop gain = O(s)/E(s)
HOL(s) = K*b*(s+a)/(a*(S+b)*s)
K = open loop gain
a = loop filter zero; ta = 1/a = zero time constant
b = loop filter pole; tb = 1/b= pole time constant
s = j*w
HCL(s) = cloosed loop gain = O(s)/I(s)
HCL(s) = HOL(s)/(1 + HOL(s))
HCL(s) = (K*b*(s+a)/a) / (s^2 + s*(b+K*b/a) + K*b)
Standard control loop form:
HCL(s) = ((s/Zero+1)*wn^2/(s^2 + s*(2*DF*wn) + wn^2)
wn = loop natural frequency (radians/sec)
DF = damping factor
Zero = open loop zero
Derived frequency and time domain characteristics:
Zero = a
ZeroN = normalized zero = a/wn
wn = sqrt(K*b)
DFactor = damping factor = (b+K*b/a)/(2*sqrt(K*b))
DFactor = (b + wn/ZeroN)/ (2*wn)
BW_CL = cloosed loop bandwidth
BW_CL = 2*pi*wn*sqrt(A + sqrt(A^2+1))
BWn_CL = closed loop noise bandwidth

BWn_CL = wn*(wn^2/Zero + Zero) / (8*Zero*DFactor)
PcntOS = percent step overshoot
PcntOS = 100*sqrt(1+ 1/ZeroN^2 -2*DFactor/ZeroN) * exp(-DFactor*Phi/sqrt(1-DFactor^2))
where
A= 1 + 1/ZeroN^2 - 2*DFactor^2
Phi = pi - arctan (sqrt(1-DFactor^2) / (ZeroN-DFactor)

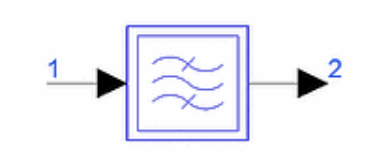
References

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. K. Feher, *Digital Communications: Microwave Applications*, Prentice-Hall, Englewood Cliffs, N.J., 1981.
4. J. G. Proakis, *Digital Communications*, McGraw-Hill, 1989.

Components

- *BPF BesselTimed* (timed)
- *BPF ButterworthTimed* (timed)
- *BPF ChebyshevTimed* (timed)
- *BPF EllipticTimed* (timed)
- *BPF GaussianTimed* (timed)
- *BPF RaisedCosineTimed* (timed)
- *CktLoopFilter110* (timed)
- *CktLoopFilter211* (timed)
- *LPF BesselTimed* (timed)
- *LPF ButterworthTimed* (timed)
- *LPF ChebyshevTimed* (timed)
- *LPF EllipticTimed* (timed)
- *LPF GaussianTimed* (timed)
- *LPF RaisedCosineTimed* (timed)
- *SP LoopFilter110* (timed)
- *SP LoopFilter211* (timed)

BPF_BesselTimed



Description: Bandpass Bessel response filter
Library: Timed, Filters
Class: TSDFBPF_BesselTimed
Derived From: basePLossStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
Loss	power loss in dB referenced to matched source and load resistors	0.0		real	[0, ∞)
FCenter	center frequency of filter passband	1000000.0	Hz	real	(0, ∞)
PassBandwidth	passband bandwidth of filter	2000.0	Hz	real	(0, ∞)
PassAtten	passband attenuation in dB, 0.01 < = PassAtten < = 3	3.		real	[0.01, 3]
StopBandwidth	stopband bandwidth, StopBandwidth > PassBandwidth, not used if N > 0	1200000	Hz	real	(PassBandwidth, ∞)
StopAtten	attenuation in dB at StopBandwith, StopAtten > PassAtten, not used if N > 0	50.		real	(3, ∞)
N	filter order, 1 < = N < = 12; the N=0 feature is not available in this product release	5		real	[0, 12]
WindowType	window applied to filter impulse response: Rectangular, Bartlett, Hanning, Hamming, Flat Top	Rectangular		enum	
ImpTime	impulse time duration, ImpTime=0 defaults to 10/(0.5*Bandwidth_at_3dB_down)	0.0	sec	real	[0, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

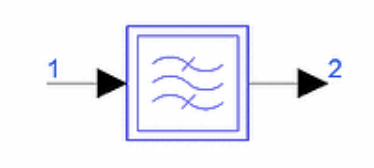
Pin Outputs

Pin	Name	Description	Signal Type
2	output	output signal	timed

Notes/Equations

1. Refer to *Timed Filters* (timed).

BPF_ButterworthTimed



Description: Bandpass Butterworth response filter
Library: Timed, Filters
Class: TSDFBPF_ButterworthTimed
Derived From: basePLossStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
Loss	power loss in dB referenced to matched source and load resistors	0.0		real	[0, ∞)
FCenter	center frequency of filter passband	1000000.0	Hz	real	(0, ∞)
PassBandwidth	passband bandwidth of filter	2000.0	Hz	real	(0, ∞)
PassAtten	passband attenuation in dB, 0.01 < = PassAtten < = 3	3.		real	[0.01, 3]
StopBandwidth	stopband bandwidth, StopBandwidth > PassBandwidth, not used if N > 0	1200000	Hz	real	(PassBandwidth, ∞)
StopAtten	attenuation in dB at StopBandwith, StopAtten > PassAtten, not used if N > 0	50.		real	(3, ∞)
N	filter order, 0 < = N < = 15; if N=0, then N is set based on StopBandwidth and StopAtten	5		real	[0, 15]
WindowType	window applied to filter impulse response: Rectangular, Bartlett, Hanning, Hamming, Flat Top	Rectangular		enum	
ImpTime	impulse time duration, ImpTime=0 defaults to 10/(0.5*Bandwidth_at_3dB_down)	0.0	sec	real	[0, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

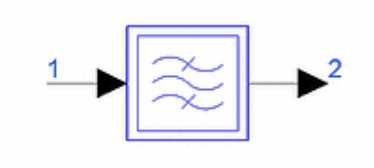
Pin Outputs

Pin	Name	Description	Signal Type
2	output	output signal	timed

Notes/Equations

1. Refer to *Timed Filters* (timed).

BPF_ChebyshevTimed



Description: Bandpass Chebyshev response filter
Library: Timed, Filters
Class: TSDFBPF_ChebyshevTimed
Derived From: basePLossStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
Loss	power loss in dB referenced to matched source and load resistors	0.0		real	[0, ∞)
FCenter	center frequency of filter passband	1000000.0	Hz	real	(0, ∞)
PassBandwidth	passband bandwidth of filter	2000.0	Hz	real	(0, ∞)
PassRipple	passband ripple in dB, 0.01 < = PassRipple < = 3	1.0		real	[0.01, 3]
StopBandwidth	stopband bandwidth, StopBandwidth > PassBandwidth, not used if N > 0	1200000	Hz	real	(PassBandwidth, ∞)
StopAtten	attenuation in dB at StopBandwith, StopAtten > PassRipple, not used if N > 0	50.		real	(3, ∞)
N	filter order, 0 < = N < = 15; if N=0, then N is set based on StopBandwidth and StopAtten	5		real	[0, 15]
WindowType	window applied to filter impulse response: Rectangular, Bartlett, Hanning, Hamming, Flat Top	Rectangular		enum	
ImpTime	impulse time duration, ImpTime=0 defaults to 10/(0.5*PassBandwidth)	0.0	sec	real	[0, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

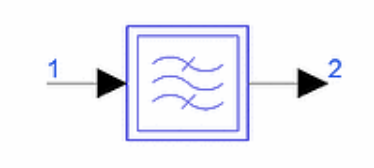
Pin Outputs

Pin	Name	Description	Signal Type
2	output	output signal	timed

Notes/Equations

1. Refer to *Timed Filters* (timed).

BPF_EllipticTimed



Description: Bandpass elliptic response filter
Library: Timed, Filters
Class: TSDFBPF_EllipticTimed
Derived From: basePLossStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
Loss	power loss in dB referenced to matched source and load resistors	0.0		real	[0, ∞)
FCenter	center frequency of filter passband	1000000.0	Hz	real	(0, ∞)
PassBandwidth	passband bandwidth of filter	2000.0	Hz	real	(0, ∞)
PassRipple	passband ripple in dB, 0.01 < = PassRipple < = 3	1.0		real	[0.01, 3]
StopBandwidth	stopband bandwidth, StopBandwidth > PassBandwidth, not used if N > 0	1200000	Hz	real	(PassBandwidth, ∞)
StopAtten	attenuation in dB at StopBandwith, StopAtten > PassRipple, not used if N > 0	50.		real	(3, ∞)
N	filter order, 0 < = N < = 12; if N=0, then N is set based on StopBandwidth and StopAtten	5		real	[0, 12]
WindowType	window applied to filter impulse response: Rectangular, Bartlett, Hanning, Hamming, Flat Top	Rectangular		enum	
ImpTime	impulse time duration, ImpTime=0 defaults to 20/(0.5*PassBandwidth)	0.0	sec	real	[0, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

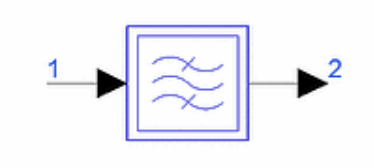
Pin Outputs

Pin	Name	Description	Signal Type
2	output	output signal	timed

Notes/Equations

1. Refer to *Timed Filters* (timed).

BPF_GaussianTimed



Description: Bandpass ideal Gaussian filter
Library: Timed, Filters
Class: TSDFBPF_GaussianTimed
Derived From: basePLossStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
Loss	power loss in dB referenced to matched source and load resistors	0.0		real	[0, ∞)
FCenter	center frequency of filter passband	1000000.0	Hz	real	(0, ∞)
PassBandwidth	passband bandwidth of filter	2000.0	Hz	real	(0, ∞)
PassAtten	attenuation in dB at PassBandwidth, 0.01 < = PassAtten < = 3	3.		real	[0.01, 3]
WindowType	window applied to filter impulse response: Rectangular, Bartlett, Hanning, Hamming, Flat Top	Rectangular		enum	
ImpTime	impulse time duration, ImpTime=0 defaults to 10/(0.5*PassBandwidth)	0.0	sec	real	[0, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

Pin Outputs

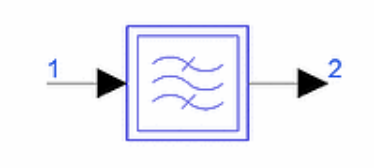
Pin	Name	Description	Signal Type
2	output	output signal	timed

Notes/Equations

1. Refer to *Timed Filters* (timed).
2. The delay introduced by the filter is approximately equal to:

$$Delay = \frac{0.6}{\sqrt{3.0/(PassAtten)} \times PassFreq}$$

BPF_RaisedCosineTimed



Description: Bandpass raised-cosine filter
Library: Timed, Filters
Class: TSDFBPF_RaisedCosineTimed
Derived From: basePLossStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
Loss	power loss in dB referenced to matched source and load resistors	0.0		real	[0, ∞)
FCenter	center frequency of filter passband	1000000.0	Hz	real	(0, ∞)
Bandwidth	passband bandwidth of filter; recommended as 1/SymbolTime	1000.0	Hz	real	(0, ∞)
ExcessBw	raised cosine filter excess bandwidth	0.5		real	[0, 1]
Type	type of raised-cosine model: Impulse model, Model with pulse equalization	Impulse model		enum	
SquareRoot	use square-root raised-cosine model: No, Yes	No		enum	
Delay	filter time delay; recommended as 4*SymbolTime	0.004	sec	real	(0, ∞)
WindowType	window applied to filter impulse response: Rectangular, Bartlett, Hanning, Hamming, Flat Top	Rectangular		enum	

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

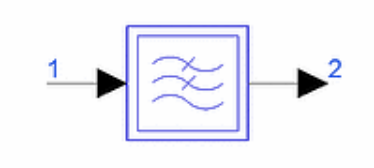
Pin Outputs

Pin	Name	Description	Signal Type
2	output	output signal	timed

Notes/Equations

1. Refer to *Timed Filters* (timed).

CktLoopFilter110



Description: Second order control loop filter, one pole, one zero, analog circuit
Library: Timed, Filters
Class: TSDFCktLoopFilter110

Parameters

Name	Description	Default	Sym	Unit	Type	Range
DampingFactor	loop damping factor	0.707	ζ		real	(0, ∞)
NormalizedZero	loop normalized zero	1	δ *		real	(0, ∞)
BW	closed loop -3dB frequency	6.5e6		Hz	real	(0, ∞)
ExternalGain	open loop gain external to this filter	7.5e6			real	(0, ∞)
C	loop filter internal capacitor value	20e-12		F	real	(0, ∞)
OpAmpGain0	loop filter internal op amp gain at DC	100000			real	(0, ∞)
MaxTimeStep	circuit transient cosimulation MaxTimeStep value	0.01e-6			real	(0, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
2	output	output signal	timed

Notes/Equations

1. This lowpass filter is for use as the loop filter in a second-order type-one feedback control loop. It contains one pole and one zero; the pole is not an integrator. It has infinite input and zero output resistance. It is modeled as an Analog Subcircuit that relies on Transient Cosimulation. Its circuit design is an op amp circuit with resistors and a capacitor that results in the frequency domain voltage transfer function represented by:

   $$\frac{K_F b(s+a)}{a(s+b)}$$

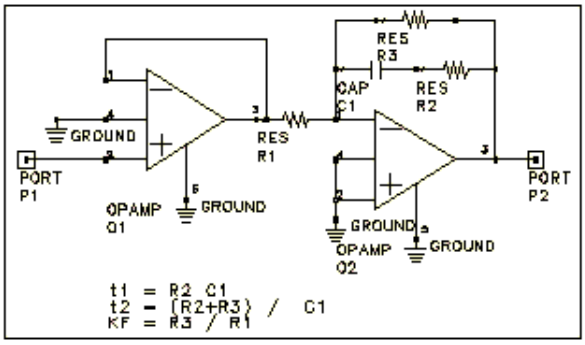
   where,
   $s = j\omega$; ω is radian frequency; j is the complex operator.
   K_F = loop filter low-frequency gain

   a = filter zero
   b = filter pole
   The expectation for second-order type-one control loop applications is that, external

to this loop filter the open loop transfer function has an additional gain factor (ExternalGain) and an integrator (the second pole). A second-order control loop has two poles. A type-one control loop has an integrator as one of its poles.

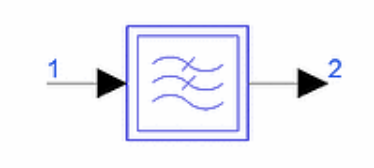
2. The design of a second-order type-one control loop may begin with various specifications in the frequency or time domain. For this discussion with example, assume the following parameters values are specified.
   ζ loop damping factor (DampingFactor): 0.707
   δ * loop normalized zero (NormalizedZero): 1.0
   BW closed loop −3 dB frequency (BW): 4.5 kHz
   From these values, we obtain the desired open- and closed-loop transfer function parameters from standard second-order control loop theory. Let the open loop transfer function be GB(s). The open- and closed-loop parameters are:
   $\omega n = 2\pi BW/\text{sqrt}(A+\text{sqrt}(A^2+1))$; $A=1+1/\delta^{*2}-2\zeta^2$
   $= 1.8197\ (10^4)$
   $a = \delta^* \omega n$
   $= 1.8197\ (10^4)$
   $b = 2\zeta w_n - \omega n^2/\delta = \omega n\ (2\zeta - 1/\delta^*)$
   $= 7.5335\ (10^3)$
   $KB = \omega n^2/b$
   $= 4.3955\ (10^4)$
   τ 1 = zero time constant 1/a
   τ 2 = pole time constant 1/b
   Since the loop filter gain constant, K_F, is multiplied by the open loop gain (ExternalGain) external to this filter to achieve the open loop gain KB, then we get:
   K_F = KB/ExternalGain

   For additional information on second order control loop theory see: *Linear Control Systems* by J. L. Melsea and D. G. Schultz, McGraw-Hill, Inc., New York, 1969.
3. The op amp circuit has the form shown here:



   First-order loop filter, one zero, no integrator, F110(s)
   R1 = (τ 1−τ 2) × ExternalGain/KB/C
   R2 = τ 1/C
   R3 = τ 2/C − R2
4. Refer to *Timed Filters* (timed).
5. This component is useful in control loops; for details, refer to *Second-Order Control Loop Filters* (timed).

CktLoopFilter211



Description: Second order control loop filter, 2 poles (with one integrator), one zero, analog circuit
Library: Timed, Filters
Class: TSDFCktLoopFilter211

Parameters

Name	Description	Default	Sym	Unit	Type	Range
DampingFactor	loop damping factor	0.707	ζ		real	(0, ∞)
NormalizedZero	loop normalized zero	1	δ *		real	(0, ∞)
BW	closed loop -3dB frequency	4.5e3		Hz	real	(0, ∞)
ExternalGain	open loop gain external to this filter	1000			real	(0, ∞)
C2	loop filter internal capacitor value	200e-12		F	real	(0, ∞)
OpAmpGain0	loop filter internal op amp gain at DC	100000			real	(0, ∞)
MaxTimeStep	circuit transient cosimulation MaxTimeStep value	1e-6			real	(0, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
2	output	output signal	timed

Notes/Equations

1. This lowpass filter is for use as the loop filter in a second-order type-one feedback control loop. It contains two poles and one zero. One of the poles is an integrator. It has infinite input and zero output resistance. It is modeled as an analog subcircuit that relies on transient cosimulation. Its circuit design is an op amp circuit with resistors and capacitors that results in the frequency domain voltage transfer function represented by:

   $$\frac{K_F b(s+a)}{a(s+b)}$$

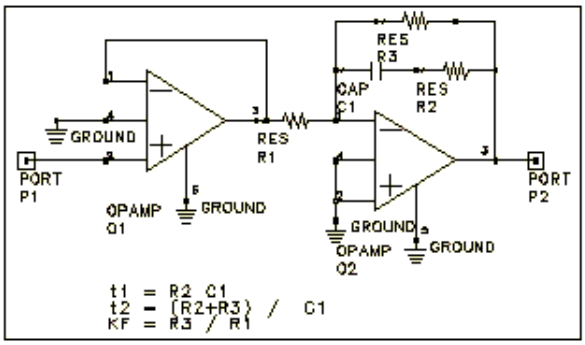
   where,
   $s = j\,\omega$; ω is radian frequency; j is the complex operator.
   K_F = loop filter low-frequency gain

   a = filter zero

b = filter pole

For second-order type-one control loop applications, the expectation is that external to this loop filter the open loop transfer function has an additional gain factor (ExternalGain). A second-order control loop has two poles. A type one control loop has an integrator as one of its poles.

2. Refer to CktLoopFilter110 for defining equations.
3. The op amp circuit has the form shown here.



Second-order loop filter, one zero, one integrator, F211(s).

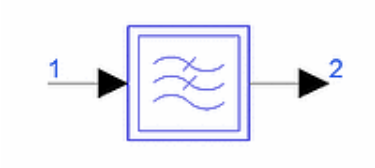
R1 = 1/C1/(KB/ExternalGain)

R2 = τ2/C2

C1 = τ1/R2 − C2

4. Refer to *Timed Filters* (timed).
5. This component is useful in control loops; for details, refer to *Second-Order Control Loop Filters* (timed).

LPF_BesselTimed



Description: Lowpass Bessel response filter
Library: Timed, Filters
Class: TSDF_LPF_BesselTimed
Derived From: basePLossStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
Loss	power loss in dB referenced to matched source and load resistors	0.0		real	[0, ∞)
PassFreq	passband edge frequency	1000000	Hz	real	(0, ∞)
PassAtten	attenuation in dB at PassFreq, 0.01 <= PassAtten <= 3	3.		real	(0, ∞)
StopFreq	stopband edge frequency, StopFreq > PassFreq, not used if N > 0	1200000	Hz	real	(PassFreq, ∞)
StopAtten	attenuation in dB at StopFreq, StopAtten > PassAtten, not used if N > 0	50.		real	(3, ∞)
N	filter order, 1 <= N <= 12; the N=0 feature is not available in this product release	5		int	[0, 12]
WindowType	window applied to filter impulse response: Rectangular, Bartlett, Hanning, Hamming, Flat Top	Rectangular		enum	
ImpTime	impulse time duration, ImpTime=0 defaults to 10/(Bandwidth_at_3dB_down)	0.0	sec	real	[0, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

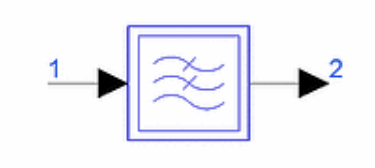
Pin Outputs

Pin	Name	Description	Signal Type
2	output	output signal	timed

Notes/Equations

1. Refer to *Timed Filters* (timed).

LPF_ButterworthTimed



Description: Lowpass Butterworth response filter
Library: Timed, Filters
Class: TSDF_LPF_ButterworthTimed
Derived From: basePLossStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
Loss	power loss in dB referenced to matched source and load resistors	0.0		real	[0, ∞)
PassFreq	passband edge frequency	1000000	Hz	real	(0, ∞)
PassAtten	attenuation in dB at PassFreq, 0.01 <= PassAtten <= 3	3.		real	(0, ∞)
StopFreq	stopband edge frequency, StopFreq > PassFreq, not used if N > 0	1200000	Hz	real	(PassFreq, ∞)
StopAtten	attenuation in dB at StopFreq, StopAtten > PassAtten, not used if N > 0	50.		real	(PassAtten, ∞)
N	filter order, 0 <= N <= 15; if N=0, then N is set based on StopFreq and StopAtten	5		int	[0, 15]
WindowType	window applied to filter impulse response: Rectangular, Bartlett, Hanning, Hamming, Flat Top	Rectangular		enum	
ImpTime	impulse time duration, ImpTime=0 defaults to 10/(Bandwidth_at_3dB_down)	0.0	sec	real	[0, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

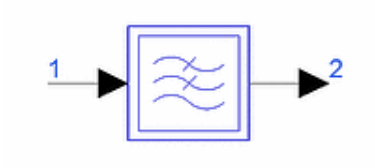
Pin Outputs

Pin	Name	Description	Signal Type
2	output	output signal	timed

Notes/Equations

1. Refer to *Timed Filters* (timed).

LPF_ChebyshevTimed



Description: Lowpass Chebyshev response filter
Library: Timed, Filters
Class: TSDF_LPF_ChebyshevTimed
Derived From: basePLossStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
Loss	power loss in dB referenced to matched source and load resistors	0.0		real	[0, ∞)
PassFreq	passband edge frequency	1000000	Hz	real	(0, ∞)
PassRipple	passband ripple in dB, 0.01 <= PassRipple <= 3	1.0		real	[0.01:3]
StopFreq	stopband edge frequency, StopFreq > PassFreq, not used if N > 0	1200000	Hz	real	[PassFreq, ∞)
StopAtten	attenuation in dB at StopFreq, StopAtten > PassAtten, not used if N > 0	50.		real	[PassRipple, ∞)
N	filter order, 0 <= N <= 15; if N=0, then N is set based on StopFreq and StopAtten	5		int	[0, 15]
WindowType	window applied to filter impulse response: Rectangular, Bartlett, Hanning, Hamming, Flat Top	Rectangular		enum	
ImpTime	Impulse time duration, ImpTime=0 defaults to 10/PassFreq	0.0	sec	real	[0, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

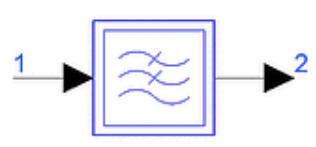
Pin Outputs

Pin	Name	Description	Signal Type
2	output	output signal	timed

Notes/Equations

1. Refer to *Timed Filters* (timed).

LPF_EllipticTimed



Description: Lowpass elliptic response filter
Library: Timed, Filters
Class: TSDF_LPF_EllipticTimed
Derived From: basePLossStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
Loss	power loss in dB referenced to matched source and load resistors	0.0		real	[0, ∞)
PassFreq	passband edge frequency	1000000	Hz	real	(0, ∞)
PassRipple	passband ripple in dB, 0.01 <= PassRipple <= 3	1.0		real	[0.01:3]
StopFreq	stopband edge frequency, StopFreq > PassFreq, not used if N > 0	1200000	Hz	real	[PassFreq, ∞)
StopAtten	attenuation in dB at StopFreq, StopAtten > PassAtten, not used if N > 0	50.		real	[PassRipple, ∞)
N	filter order, 0 <= N <= 12; if N=0, then N is set based on StopFreq and StopAtten	5		int	[0, 12]
WindowType	window applied to filter impulse response: Rectangular, Bartlett, Hanning, Hamming, Flat Top	Rectangular		enum	
ImpTime	impulse time duration, ImpTime=0 defaults to 20/PassFreq	0.0	sec	real	[0, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

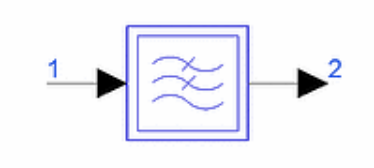
Pin Outputs

Pin	Name	Description	Signal Type
2	output	output signal	timed

Notes/Equations

1. Refer to *Timed Filters* (timed).

LPF_GaussianTimed



Description: Lowpass ideal Gaussian filter
Library: Timed, Filters
Class: TSDF_LPF_GaussianTimed
Derived From: baseLPFilter

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
Loss	power loss in dB referenced to matched source and load resistors	0.0		real	[0, ∞)
PassFreq	passband edge frequency	1000000	Hz	real	(0, ∞)
PassAtten	attenuation in dB at PassFreq, 0.01 <= PassAtten <= 3	3.0		real	[0.01, 3]
WindowType	window applied to filter impulse response: Rectangular, Bartlett, Hanning, Hamming, Flat Top	Rectangular		enum	
ImpTime	impulse time duration, ImpTime=0 defaults to 10/PassFreq	0.0	sec	real	[0, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

Pin Outputs

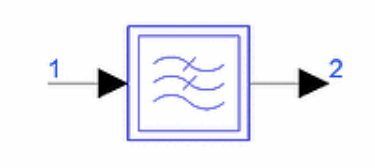
Pin	Name	Description	Signal Type
2	output	output signal	timed

Notes/Equations

1. Refer to *Timed Filters* (timed).
2. The delay introduced by the filter is approximately equal to:

$$Delay = \frac{0.6}{\sqrt{3.0/(PassAtten)} \times PassFreq}$$

LPF_RaisedCosineTimed



Description: Lowpass raised-cosine filter
Library: Timed, Filters
Class: TSDF_LPF_RaisedCosineTimed
Derived From: baseLPFilter

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
Loss	power loss in dB referenced to matched source and load resistors	0.0		real	[0, ∞)
CornerFreq	corner frequency; recommended as 1/(2*SymbolTime)	500	Hz	real	(0, ∞)
ExcessBw	raised cosine filter excess bandwidth	0.5		real	[0, 1]
Type	type of raised-cosine model: Impulse model, Model with pulse equalization	Impulse model		enum	
SquareRoot	use square-root raised-cosine model: No, Yes	No		enum	
Delay	filter time delay; recommended as 4*SymbolTime	0.004	sec	real	(0, ∞)
WindowType	window applied to filter impulse response: Rectangular, Bartlett, Hanning, Hamming, Flat Top	Rectangular		enum	

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

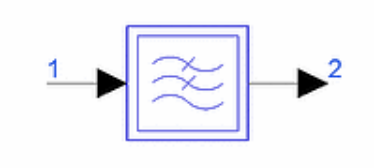
Pin Outputs

Pin	Name	Description	Signal Type
2	output	output signal	timed

Notes/Equations

1. Refer to *Timed Filters* (timed).

SP_LoopFilter110



Description: Second order control loop filter, one pole, one zero, signal processing
Library: Timed, Filters
Class: TSDFSP_LoopFilter110
Derived From: baseOmniSysStar

Parameters

Name	Description	Default	Sym	Unit	Type	Range
RIn	input resistance	DefaultRIn		Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut		Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp		Celsius	real	[-273.15, ∞)
DampingFactor	loop damping factor	0.707	ζ		real	(0, ∞)
NormalizedZero	loop normalized zero	1	δ *		real	(0, ∞)
BW	closed loop -3dB frequency	6.5e6		Hz	real	(0, ∞)
ExternalGain	open loop gain external to this filter	7.5e6			real	(0, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

Pin Ouptuts

Pin	Name	Description	Signal Type
2	output	output signal	timed

Notes/Equations

1. This lowpass filter is for use as the loop filter in a second-order type-one feedback control loop. It contains one pole and one zero. The pole is not an integrator. Its frequency domain voltage transfer function (into matched source and load resistors) is represented by:

   $$\frac{K_F b(s+a)}{a(s+b)}$$

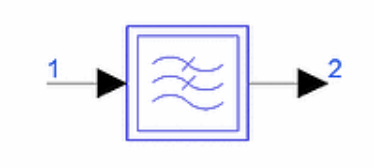
   where,
   $s = j\,\omega$; ω is radian frequency; j is the complex operator
   K_F = loop filter low-frequency gain

   a = filter zero
   b = filter pole
   The expectation for second-order type-one control loop applications is that, external

to this loop filter the open loop transfer function has an additional gain factor (ExternalGain) and an integrator (the second pole). A second-order control loop has two poles. A type-one control loop has an integrator as one of its poles.

2. The design of a second-order type-one control loop may begin with various specifications in the frequency or time domain. For this discussion with example, assume the following parameters values are specified.
   ζ loop damping factor (DampingFactor): 0.707
   δ * loop normalized zero (NormalizedZero): 1.0
   BW closed loop −3 dB frequency (BW): 4.5 kHz
   From these values, we obtain the desired open- and closed-loop transfer function parameters from standard second-order control loop theory. Let the open-loop transfer function be GB(s). The open- and closed-loop parameters are:
   $\omega n = 2\pi BW/\mathrm{sqrt}(A+\mathrm{sqrt}(A^2+1)); A=1+1/\delta^{*2}-2\zeta^2$
   $= 1.8197 (10^4)$
   $a = \delta * \omega n$
   $= 1.8197 (10^4)$
   $b = 2\zeta w_n - \omega n^2/\delta = \omega n (2\zeta - 1/\delta^*)$
   $= 7.5335 (10^3)$
   $KB = \omega n^2/b$
   $= 4.3955 (10^4)$
   τ_1 = zero time constant 1/a
   τ_2 = pole time constant 1/b
   Since, the loop filter gain constant, K_F, is multiplied by the open loop gain (ExternalGain) external to this filter to achieve open loop gain KB, then we get:
   K_F = KB/ExternalGain
3. Refer to the *Timed Filters* (timed).
4. This component is useful in control loops; for details, refer to *Second-Order Control Loop Filters* (timed).
5. For additional information on second-order control loop theory refer to *Linear Control Systems* by J. L. Melsea and D. G. Schultz, McGraw-Hill, Inc., New York, 1969.

SP_LoopFilter211



Description: Second order control loop filter, 2 poles (with one integrator), one zero, signal processing
Library: Timed, Filters
Class: TSDFSP_LoopFilter211
Derived From: baseOmniSysStar

Parameters

Name	Description	Default	Sym	Unit	Type	Range
RIn	input resistance	DefaultRIn		Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut		Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp		Celsius	real	[-273.15, ∞)
DampingFactor	loop damping factor	0.707	ζ		real	(0, ∞)
NormalizedZero	loop normalized zero	1	δ *		real	(0, ∞)
BW	closed loop -3dB frequency	4.5e3		Hz	real	(0, ∞)
ExternalGain	open loop gain external to this filter	1000			real	(0, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
2	output	output signal	timed

Notes/Equations

1. This lowpass filter is for use as the loop filter in a second-order type-one feedback control loop. It contains two poles and one zero; one of the poles is an integrator. Its frequency domain voltage transfer function (into matched source and load resistors) is represented by:

$$\frac{K_F b(s+a)}{as(s+b)}$$

where,
$s = j\,\omega$; ω is radian frequency; j is the complex operator.
K_F = loop filter low-frequency gain

a = filter zero
b = filter pole

For second-order type-one control loop applications, the expectation is that external to this loop filter the open loop transfer function has an additional gain factor (ExternalGain). A second-order control loop has two poles. A type one control loop has an integrator as one of its poles.

2. Refer to *SP_LoopFilter110* (timed) for defining equations.
3. Refer to *Timed Filters* (timed).
4. This component is useful in control loops; for details, refer to *Second-Order Control Loop Filters* (timed).

Timed Linear Components

The Timed Linear library contains time domain linear components that include phase shifter, time delay, signal summer, signal splitter, and switches.

Each timed linear component consumes and produces baseband or RF (complex envelope) timed signals. If a component receives another class of scalar signal, the received signal is automatically converted to a timed (baseband) type of signal. Auto conversion from the complex scalar to the timed class is not allowed. The user must explicitly use the CxToTimed signal converter for this transformation. These components do not accept any matrix class of signal.

Some components will accept or produce a baseband or RF (complex envelope) timed signal only. If a component requires an RF (complex envelope) timed signal, then its receipt of a baseband timed signal will declare an error and the simulation will stop. If a baseband timed signal is the required input, then any received RF (complex envelope) timed signal is first transformed into its baseband equivalent form before use by the component.

An RF (complex envelope) timed signal is converted to its equivalent baseband form:

$$V_{bb}(t) = Re\left\{(v_{RF}(t))e^{j2\pi f_c t}\right\} = Re\left\{(v_I(t) + jv_Q(t))e^{j2\pi f_c t}\right\}$$

where
$V_{bb}(t)$ is the total representation for the RF signal (also called the baseband representation)
$v_{RF}(t)$ is the RF signal complex envelope at characterization frequency fc (also called the equivalent complex baseband envelope representation for the RF signal)
$v_I(t)$ is the RF timed signal in-phase envelope

$v_Q(t)$ is the RF timed signal quadrature-phase envelope

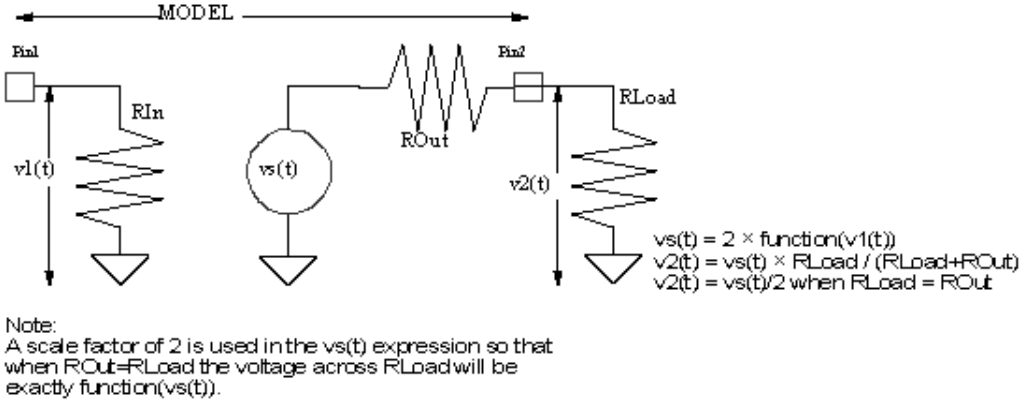
f_c is the RF signal characterization frequency

For this equivalence to be valid, the simulation time step must be less than the inverse of the characterization frequency and the RF signal information content has an information bandwidth less than the RF characterization frequency.

All linear components have user-specified input (RIn) and output (ROut) resistance. Input resistance is for a shunt resistor at each component input pin; output resistance is for a series resistor at each component output pin.

Each component internal output signal vs(t), has a value equal to twice the output pin signal v2(t), when the output is connected to a matched resistor load. The circuit model illustrated in 1 Input, 1 Output Circuit Model for a 2-port (1 input, 1 output) timed linear component demonstrates this relationship.

1 Input, 1 Output Circuit Model



The output pin signal, v2(t), at the output series resistance is dependent on the value of the load resistance connected to it. When the load resistor RLoad, is equal to the model output resistor ROut, the value of v2(t) is equal to vs(t)/2; otherwise, based on the voltage divider action, v2(t) is:
$v\,2\,(t) = vs(t) \times RLoad/(RLoad + ROut\ )$

The input and output resistor values must be greater than 0 ohm.

The input and output resistors contribute additive thermal noise power (kTB) to the output signal when the specified resistance temperature (RTemp) is greater than absolute zero (−273.15°C) where:
k = Boltzmann's constant
T = temperature in Kelvin
B = simulation frequency bandwidth:
1/2/tstep if signal is a timed baseband signal;
1/tstep if signal is a timed complex envelope signal

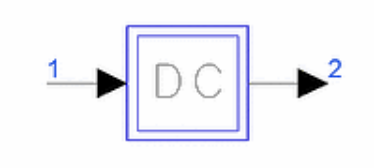
When RTemp > −273.15, the noise contributed from each resistor instance is an independent noise process. This noise is dependent on the value of DefaultSeed in the DF (data flow) controller. When DefaultSeed=0, the noise generated for each simulation is different. When DefaultSeed>0, the noise generated for each simulation, though random, has the same initial seed starting condition and thus results in reproducible noise for each simulation.

Note
Information regarding time domain signal differences between ADS Ptolemy simulations and Circuit Envelope and Transient simulations is given in the *Timed Synchronous Dataflow* (ptolemy).

Components

- *DcShift* (timed)
- *DelayRF* (timed)
- *DSampleRF* (timed)
- *FcChange* (timed)
- *FcTrackSecondary* (timed)
- *IntDumpTimed* (timed)
- *MatchedLoss* (timed)
- *PhaseShiftRF* (timed)
- *RFtoCWRF* (timed)
- *SampleAndHold* (timed)
- *SampleAndHoldRF* (timed)
- *SBlock* (timed)
- *SplitterRF* (timed)
- *SummerRF* (timed)
- *SwitchSPDT* (timed)
- *SwitchSPST* (timed)
- *USampleRF* (timed)
- *VcDelayRF* (timed)
- *VcPhaseShiftRF* (timed)

DcShift



Description: DC shift
Library: Timed, Linear
Class: TSDF_DcShift
Derived From: baseStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
Vdc	voltage level offset added to input signal	0	V	real	(∞, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
2	output	output signal	timed

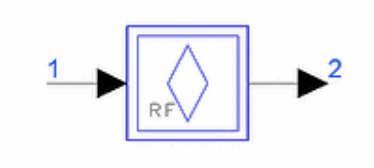
Notes/Equations

1. V_2 (t) = V_1 (t) + Vdc. The output signal is always represented in terms of its inphase and quadrature components about a frequency of 0 Hz, that is,

   $$V_2(t) = Re\left\{(v_{I2}(t) + jv_{Q2})e^{j2\pi f_{c2}t}\right\}$$

   where
   v_{I2} (t) = V_1 (t) + Vdc
   v_{Q2} (t) = 0
   f_{c2} = 0
2. For information regarding timed linear component signals, refer to *Timed Linear Components* (timed).

DelayRF



Description: Time delay
Library: Timed, Linear
Class: TSDF_DelayRF
Derived From: baseOmniSysStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
Delay	time delay, must be >= Tstep; set to -1 for one TStep delay	-1	sec	real	{-1} or [TStep, ∞)†
InterpolationMethod	signal interpolation method: none, linear	none		enum	
IncludeCarrierPhaseShift	include RF carrier phase shift: No, Yes	Yes		enum	

† TStep is the simulation time step for the component input signal; the -1 value will set Delay to TStep.

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
2	output	output signal	timed

Notes/Equations

1. DelayRF delays the input signal by *Delay*. When the input is an RF (complex envelope) signal, the delay is applied to the RF envelope and optionally to the RF carrier phase (when *IncludeCarrierPhaseShift* is set to *Yes*) as described in the following equations.
   The input signal $V_1(t)$ is represented by its inphase and quadrature components at its carrier frequency f_{c1}

$$V_1(t) = Re\left\{v_1(t)e^{j2\pi f_{c1}t}\right\}, v_1(t) = v_{I1}(t) + j\, v_{Q1}(t)$$

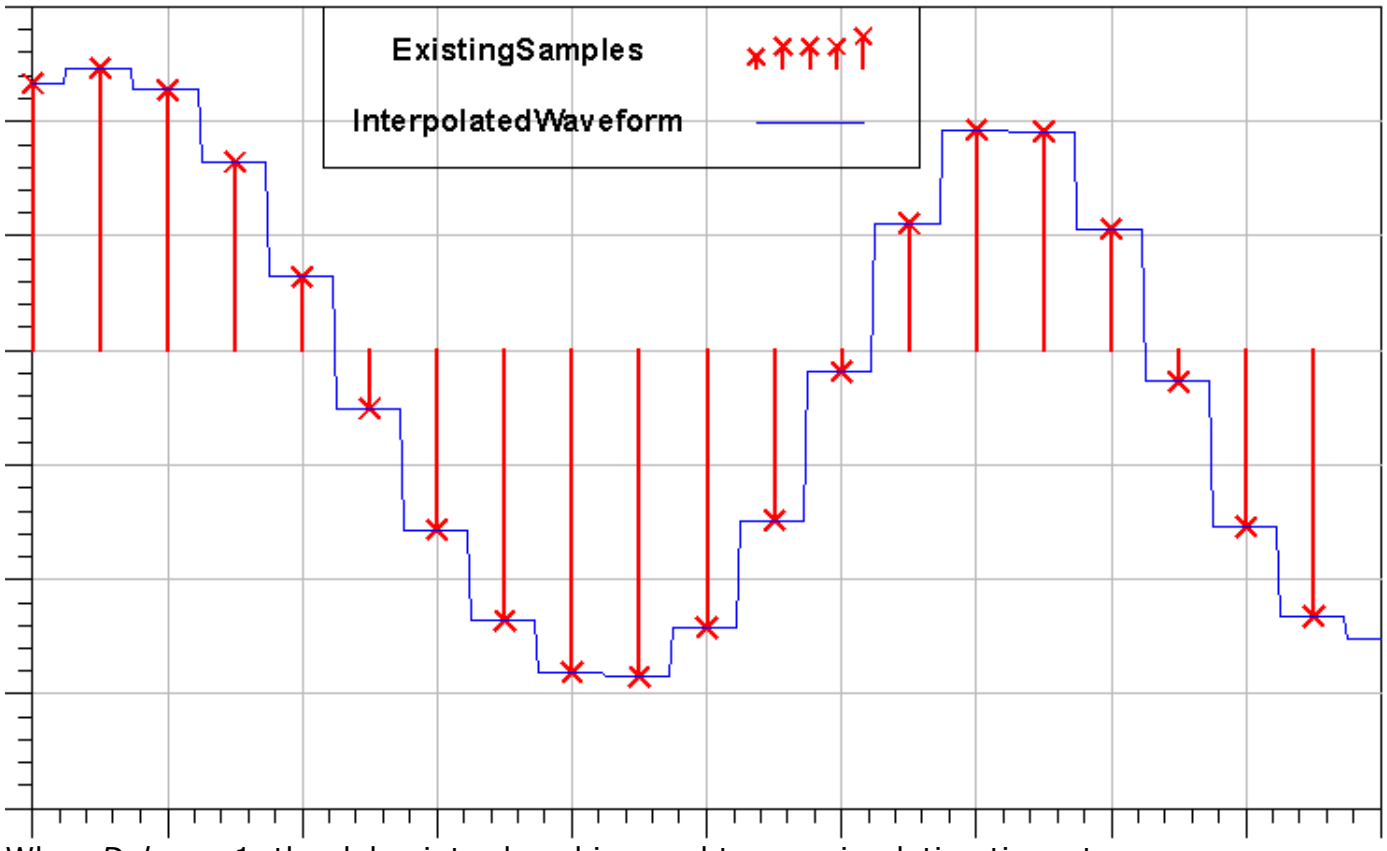
For a real baseband signal f_{c1} = 0 and $v_{Q1}(t)$ = 0.

The output signal $V_2(t)$ is represented by its inphase and quadrature components at its carrier frequency f_{c2}

$$V_2(t) = Re\left\{v_2(t)e^{j2\pi f_{c2}t}\right\}$$

where
$f_{c2} = f_{c1}$
$v_2(t) = v_1(t - Delay)$, when *IncludeCarrierPhaseShift* is set to *No*
$v_2(t) = v_1(t - Delay)\cdot e^{-j2\pi\cdot f_{c2}\cdot Delay}$, when *IncludeCarrierPhaseShift* is set to *Yes*

When *InterpolationMethod* is set to *linear* or *Lagrange*, $v_1(t - Delay)$ is found using the corresponding interpolation method.
When *InterpolationMethod* is set to *none*, $v_1(t - Delay)$ is found using a 0^{th}-order interpolation method that maps $v_1(t - Delay)$ to the closest existing sample (see graph below).

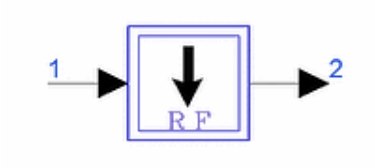


2. When *Delay*=−1, the delay introduced is equal to one simulation time step.
3. When *DelayRF* is used in feedback loops in multirate systems, it may cause deadlocks to be reported even though the Delay is set to multiple simulation time steps. In this

case, *DelayRF* should be replaced by a *DelayRF* with a *Delay* of 1 simulation time step, and the additional delay must be provided using the *Delay* (numeric) component (Numeric Control palette).

4. For information regarding timed linear component signals, refer to *Timed Linear Components* (timed).

DSampleRF



Description: Down sampler for time domain signals
Library: Timed, Linear
Class: TSDF_DSampleRF
Derived From: baseStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
Ratio	downsampling ratio	2		int	[1, ∞)
Phase	downsampling phase	0		int	[0, Ratio-1]
AntiAliasingFilter	turn off/on anti-aliasing filter before downsampling: Off, On	Off		enum	
ExcessBW	excess bandwidth of raised cosine anti-aliasing filter	0.5		real	[0, 1]

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
2	output	output signal	timed

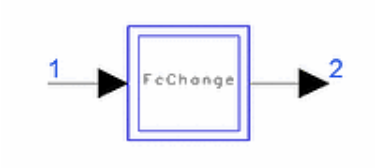
Notes/Equations

1. This component downsamples an input timed signal to produce an output timed signal sampled with a time step that is Ratio times the input time step. For every Ratio input samples consumed, one output sample is produced.
2. The Phase parameter identifies which Ratio input sample is to be used as the output sample.
   $v_2(n \times TStepOut) = v_1(n \times TStepOut + (Ratio - Phase - 1) \times TStepIn)$
   where TStepOut = Ratio×TStepIn; TStepIn is the input signal time step, and n is the output sample number
3. The AntiAliasingFilter parameter can be used to activate/de-activate an anti-aliasing filter before downsampling. When the input signal is baseband, the anti-aliasing filter is a lowpass raised-cosine filter with bandwidth equal to 1/(2 × TStepOut). When the input signal is a complex envelope signal, the anti-aliasing filter is a bandpass raised-

cosine filter with bandwidth equal to 1/TStepOut. In both cases the filter has 20 × Ratio taps (it introduces a delay of 10 × TStepOut) and its excess bandwidth can be set by the ExcessBW parameter.

4. To downsample a signal by a non-integer factor, a cascade of an upsampler and downsampler is needed. For example, to change the sampling rate of a signal from 73MHz (TStep = 13.69863 nsec) to 40MHz (TStep = 25 nsec), first pass it through the USampleRF component (Ratio = 40, Type=PolyPhaseFilter or Linear) and then through the DSampleRF component (Ratio=73). To improve simulation speed, make sure the two ratios are relatively prime; if not, divide them with their GCD (greatest common divisor). For example, if in the previous case the original sampling rate was 72 MHz, the Ratio of the USampleRF component can be set to 5 (=40/8) and the Ratio of the DSampleRF component can be set to 9 (=72/8); GCD (72, 40) = 8.
5. For information regarding timed linear component signals, refer to *Timed Linear Components* (timed).

FcChange



Description: Signal characterization frequency change
Library: Timed, Linear
Class: TSDF_FcChange
Derived From: baseStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
FcOut	new characterization frequency	0	Hz	real	[0, ∞)
Bandwidth	bandwidth of bandpass filter centered at FcOut (used when input fc=0)	0	Hz	real	[0, ∞)†

† A 0 value will set Bandwidth to FcOut

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
2	output	output signal	timed

Notes/Equations

1. This component characterizes the input signal with I and Q envelopes at a carrier frequency of f_{c1} into an equivalent representation at the new carrier frequency of f_{c2} =FcOut. The basic assumption is that the input signal is a bandpass signal; that is, it has no significant energy around 0 Hz.
2. The following equations describe how the signal is represented in Ptolemy. The use of an FcChange component will not change the nature of the signal itself; it only changes the signal's representation in the simulator.
   Consider the following example: a modulated spectrum centered around 10.5 MHz (Fc=10.5 MHz) is connected to the input of an FcChange with FcOut=10.5 MHz. The modulated spectrum at the input and output of the FcChange is observed using two SpectrumAnalzyer sinks; each sink shows a spectrum centered around 10.5 MHz.

(This can be confirmed using a TimedToData component.)

Note Warning messages will be displayed if the input signal simulation time step is too large to properly represent the output signal. A smaller value for the simulation time step is recommended when this happens.

Let Fc=FcOut. Two distinct cases must be considered.

Case 1: $f_{c1} > 0$

The input signal $V_1(t)$ at pin 1 is represented by its inphase and quadrature components about its carrier frequency:

$$V_1(t) = Re\left\{v_1(t)e^{j2\pi f_{c1}t}\right\},\ v_1(t) = v_{I1}(t) + jv_{Q1}(t)$$

The output signal $V_2(t)$ at pin 2 can be expressed as

$$V_2(t) = Re\{v_2(t)e^{j2\pi Fct}\}$$

where

$$v_2(t) = v_1(t)e^{j2\pi(f_{c1}-Fc)t}$$

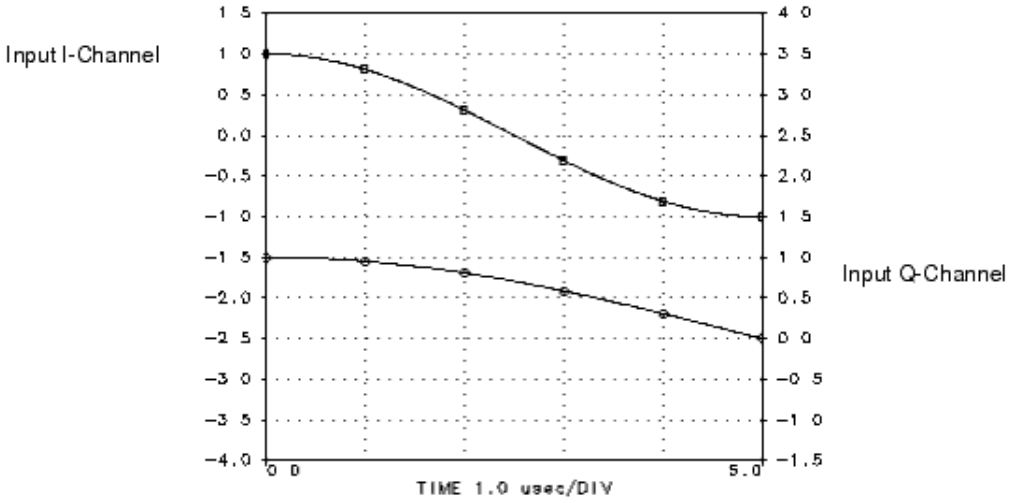
(Note that in this case the parameter Bandwidth has no effect on the output.)

Case 2: $f_{c1} = 0$

The assumption is made that the input signal is a bandpass signal with no significant energy at 0Hz and energy of interest located at FcOut±(Bandwidth/2). The inphase and quadrature phase signals are then extracted by multiplying the input signal by cos(2π Fct) and (-1) sin(2π Fct) and filtering the resulting signals with a lowpass filter of bandwidth Bandwidth (the default value of Bandwidth is FcOut, but this can be changed by the user). (Note that a delay is introduced in this case because of the filtering.)

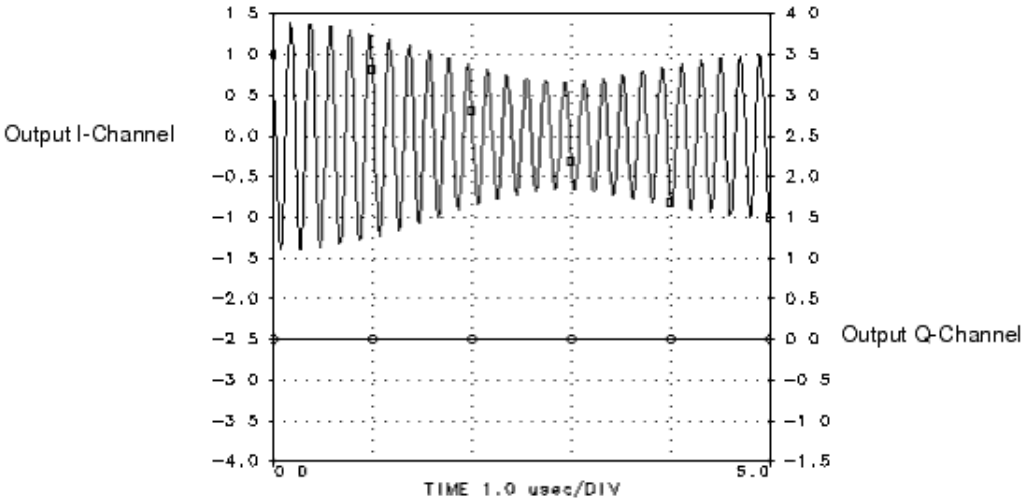
3. FcChange Component Input I- and Q-Channels, Fc1=5 MHz through FcChange Component Output I- and Q-Channels, Fc2=2.5 MHz show the performance of the FcChange component for different values of the parameter FcOut. The input signal is a QAM source with a carrier frequency at 5 MHz with the I-channel modulating signal at 100 kHz and Q-channel modulating signal at 50 kHz.
   Note the differences in the output I- and Q-signal components at different output characterization frequencies.

FcChange Component Input I- and Q-Channels,



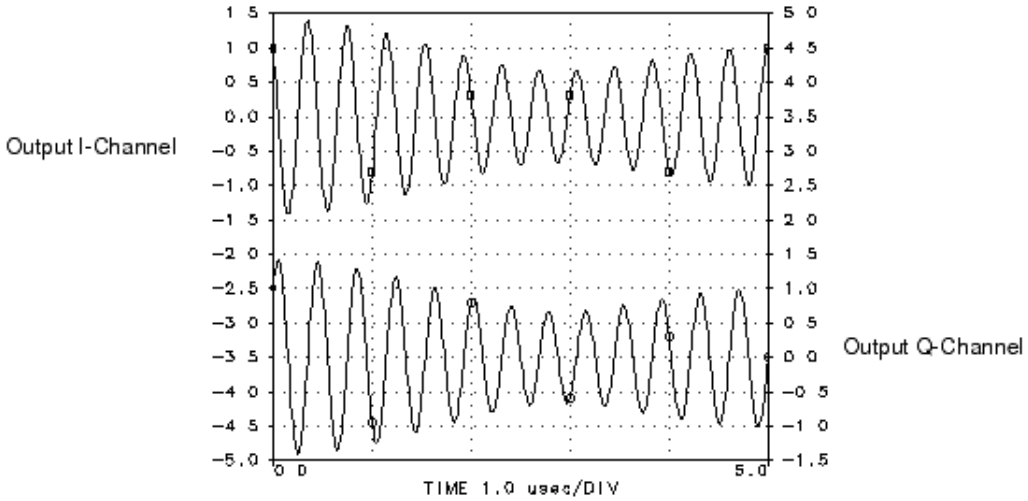
Fc1=5 MHz

FcChange Component Output I- and Q-Channels,



Fc2=0 Hz

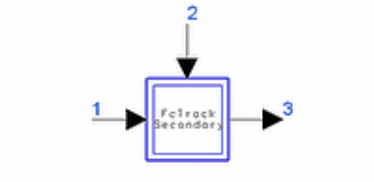
FcChange Component Output I- and Q-Channels,



Fc2=2.5 MHz

4. For information regarding timed linear component signals, refer to *Timed Linear Components* (timed).

FcTrackSecondary



Description: Track input signal characterization frequency
Library: Timed, Linear
Class: TSDF_FcTrackSecondary
Derived From: _FcChange

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
Bandwidth	bandwidth of bandpass filter centered at FcOut (used when input fc=0)	0	Hz	real	[0, ∞)†

† A 0 value will set Bandwidth to the characterization frequency of the signal at pin 2.

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed
2	fc_input	signal whose fc will be used	timed

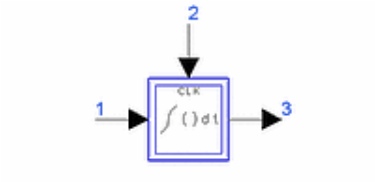
Pin Outputs

Pin	Name	Description	Signal Type
3	output	output signal	timed

Notes/Equations

1. This component characterizes the input signal with I and Q envelopes at a carrier frequency of f_{c1} into an equivalent representation at the new carrier frequency of fc3 = fc2. That is, the output signal characterization frequency tracks the input characterization frequency at pin 2. The basic assumption is that the input signal is a bandpass signal; that is, it has no significant energy around 0 Hz.
2. This component performs a signal transformation operation the same as for FcChange; for details, refer to that component documentation.
3. For information regarding timed linear component signals, refer to *Timed Linear Components* (timed).

IntDumpTimed



Description: Integrate and dump with reset
Library: Timed, Linear
Class: TSDF_IntDumpTimed
Derived From: baseStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed
2	clock	clock signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
3	output	output signal	timed

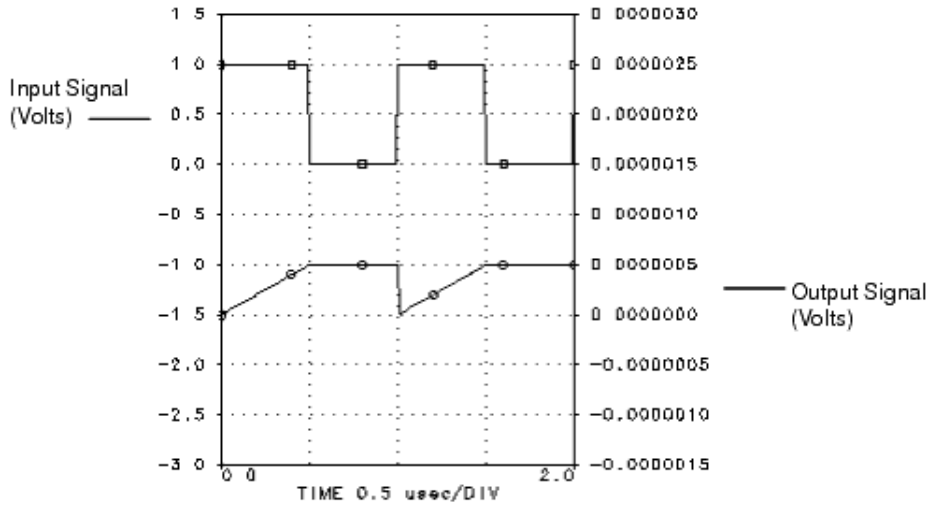
Notes/Equations

1. This component performs an integrate and dump function on the input signal $V_1(t)$, where the time of integration is determined by the clock signal $V_2(t)$. Let T_0, T_1, T_2, ... be the time instances when the positive edges of the input clock, $V_2(t)$, occur (a positive edge occurs at the instant when the clock voltage, $V_2(t)$, crosses a threshold of 0.5V). The output signal $V_3(t)$ is then determined by the following equations.

$$V_3(t) = \int_{T_k}^{t} V_1 t\,dt \qquad T_k < t \le T_{k+1}$$

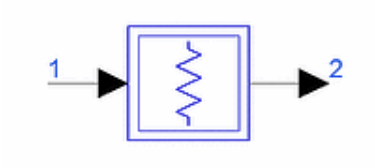
Integration is performed using the trapezoidal rule.
2. The input signal and output signal voltages of the IntDumpTimed component, with its CLK pin tied to a clock source with period 1.0 µsec, is shown below.

Input and Output Signals



3. For information regarding timed linear component signals, refer to *Timed Linear Components* (timed).

MatchedLoss



Description: Power loss referenced to matched source and load resistors
Library: Timed, Linear
Class: TSDF_MatchedLoss
Derived From: basePLossStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
Loss	power loss in dB referenced to matched source and load resistors	0.0		real	[0, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
2	output	output signal after power loss	timed

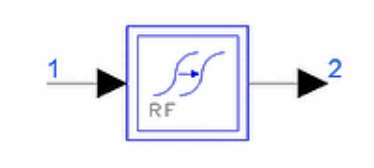
Notes/Equations

1. The output signal level is based on the input signal power loss specified in dB referenced to matched source and load resistors.
2. Loss contributes additive thermal noise power ($kT_e B$) to the output signal when the specified resistance temperature (RTemp) is greater than absolute zero (−273.15 °C) and Loss (in dB) is greater than 0, where:
   k = Boltzmann's constant
   T_e = equivalent noise temperature in Kelvin = (1 - ploss) (RTemp +273.15)
   B = simulation frequency bandwidth:
   1/2/tstep if signal is a timed baseband signal;
   1/tstep if signal is a timed complex envelope signal
   ploss = $10^{(-abs(Loss)/10)}$
   When RTemp > −273.15, the noise contributed by the Loss is an independent noise process. This noise is dependent on the value of DefaultSeed in the DF (data flow) controller. When DefaultSeed=0, then the noise generated for each simulation is different. When DefaultSeed>0, then the noise generated for each simulation, though random, has the same initial seed starting condition and thus results in reproducible

noise for each simulation.

3. For information regarding timed linear component signals, refer to *Timed Linear Components* (timed).

PhaseShiftRF



Description: RF phase shifter continuously interpolated between time steps
Library: Timed, Linear
Class: TSDF_PhaseShiftRF
Derived From: baseStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
PhaseShift	phase shift angle in degrees	0	deg	real	(-∞, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
2	output	output signal	timed

Notes/Equations

1. This component shifts the phase of the input signal by *A* degrees. The input signal must be an RF (complex envelope) signal represented by its inphase and quadrature components. The input signal at pin 1 is represented by its inphase and quadrature components about its carrier frequency

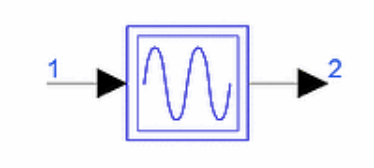
   $$V_1(t) = Re[v_1(t)e^{j2\pi f_{c1}t}], v_1(t) = v_{I1}(t) + jv_{Q1}(t)$$

   The output signal at pin 2 is given by the equation

   $$V_2(t) = Re[v_1(t)e^{jA}\ e^{j2\pi f_{c1}t}]$$

   If the input is a baseband signal ($f_{c1} = 0$) an error will be declared and simulation will stop.
2. For information regarding timed linear component signals, refer to *Timed Linear Components* (timed).

RFtoCWRF



Description: CW RF signal derived from input RF carrier frequency
Library: Timed, Linear
Class: TSDF_RFtoCWRF
Derived From: baseStar

Parameters

Name	Description	Default	Symbol	Unit	Type	Range
RIn	input resistance	DefaultRIn		Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut		Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp		Celsius	real	[-273.15, ∞)
Amplitude	envelope voltage amplitude of output continuous wave (CW) RF signal	1.0	A	V	real	[0, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input RF signal	timed

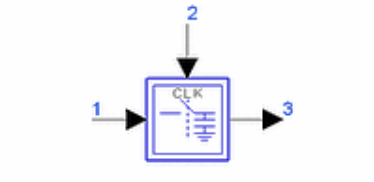
Pin Outputs

Pin	Name	Description	Signal Type
2	output	output CW RF signal	timed

Notes/Equations

1. The output signal is set to the specified amplitude at the same characterization frequency as the input signal.
   - For an input RF signal, the output is $y(t) = A \cos(2\pi F_c t)$.
   - For an input baseband signal, the output is $y(t) = A$.
2. For information regarding timed linear component signals, refer to *Timed Linear Components* (timed).

SampleAndHold



Description: Sample and hold with output decay
Library: Timed, Linear
Class: TSDF_SampleAndHold
Derived From: baseOmniSysStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
DroopRate	output voltage decay rate in fractional volts per second	0.1		real	[0, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed
2	clock	sampling clock	timed

Pin Outputs

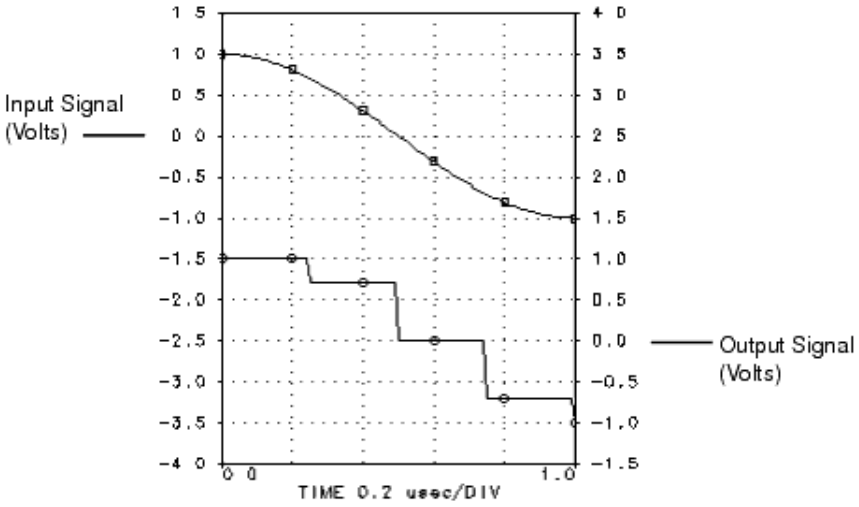
Pin	Name	Description	Signal Type
3	output	output signal	timed

Notes/Equations

1. The two inputs to this component are the input signal $V_1(t)$ and a clock signal $V_2(t)$.

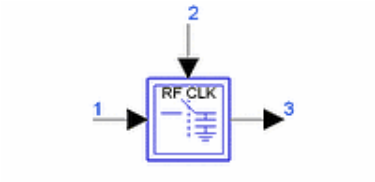
   The input signal is sampled at each rising edge of the clock signal (the input is sampled at the instant when $V_2(t)$ crosses a threshold of 0.5V). In the hold state of the sample and hold, the output voltage decays at a constant rate determined by the DroopRate parameter.
2. The output signal is always a baseband signal.
3. The input signal and output signal voltages of SampleAndHold, with clock pin tied to a clock source with period 0.25 µsec and DroopRate=0, are shown below.

SampleHold Signal Plot



4. For information regarding timed linear component signals, refer to *Timed Linear Components* (timed).

SampleAndHoldRF



Description: Sample-and-Hold with RF clock
Library: Timed, Linear
Class: TSDF_SampleAndHoldRF
Derived From: _SampleAndHold

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
DroopRate	output voltage decay rate in fractional volts per second	0.1		real	[0, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed
2	clock	sampling clock	timed

Pin Outputs

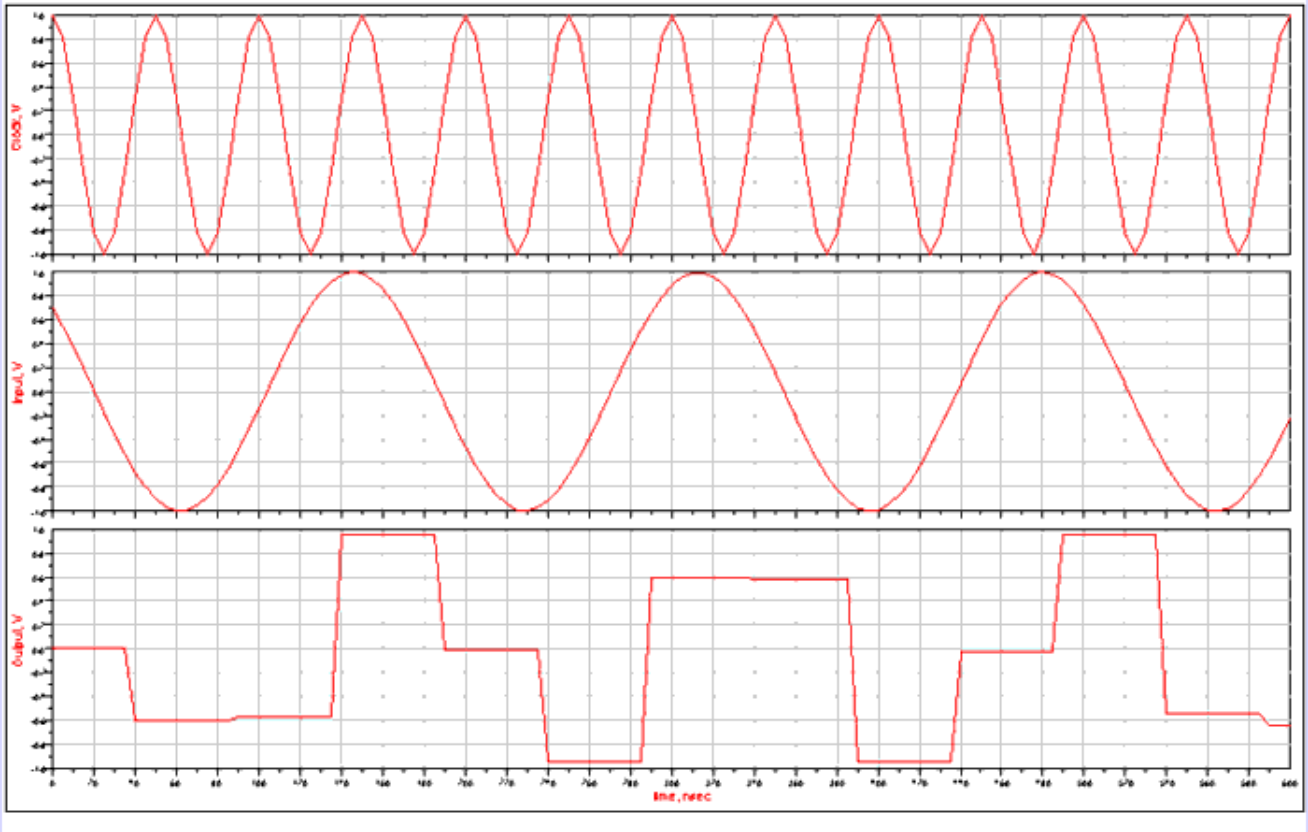
Pin	Name	Description	Signal Type
3	output	output signal	timed

Notes/Equations

1. The inputs to this component are the input signal $V_1(t)$ and an RF clock signal $V_2(t)$. The input signal is sampled at the rising zero crossing of the RF clock signal (the input is sampled at the instant when $V_2(t)$ crosses a threshold of 0V). In the hold state of the sample and hold, the output voltage decays at a constant rate determined by the parameter DroopRate.
2. The output signal is always a baseband signal.
3. This component is preferred for use compared to the *SampleAndHold* (timed) component when the input signal is desired to be sampled at the specific zero crossings of the RF Clock regardless of the simulation time step used. The model will interpolate the RF Clock signal to determine the actual time of zero crossing and use this time instant to interpolate the input signal as needed to obtain the desired sample output signal. In contrast to this operation, the *SampleAndHold* (timed) model will only sample the input signal at a simulation time step instant and with no interpolation applied between simulation time steps.
4. Input and output signal voltages of SampleAndHoldRF, with the RF clock signal set to a 20 MHz sinusoid and DroopRate=0, are shown in SampleHold Signal Plot. The RF

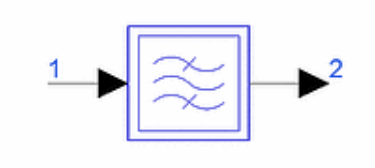
Clock signal displayed is shown in its baseband form so that upward zero crossings are visible. The input signal is a frequency modulated RF signal also shown in its baseband form so that its full signal detail is visible. The output is as simulated.

SampleHold Signal Plot



5. For information regarding timed linear component signals, refer to *Timed Linear Components* (timed).

SBlock



Description: Read S21 Data
Library: Timed, Linear
Class: TSDFSBlock
Derived From: baseTFIR

Parameters

Name	Description	Default	Unit	Type	Range
File	File name for S21 data	myfile.s2p		filename	
Type	File type: Dataset, Touchstone, CITIfile	Touchstone		enum	
InterpMode	Interpolation mode: Linear, Cubic Spline, Cubic	Linear		enum	
N	Number of points for impulse response	128		int	[2, ∞)
RIn	Input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	Output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	signalIn	input	timed

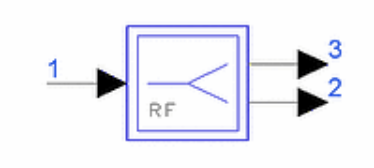
Pin Outputs

Pin	Name	Description	Signal Type
2	signalOut	output	timed

Notes/Equations

1. This component is intended for modeling measured/simulated S-parameter data for filters. Refer to *Timed Filters* (timed) for the general filter modeling method.
2. SBlock inherits an Fc parameter from the component preceding it. If the preceding component has Fc=0, SBlock is characterized as a lowpass or highpass filter with real FIR tap coefficients. If the preceding component has Fc>0, SBlock is characterized as a bandpass or bandstop filter with complex FIR tap coefficients.
3. SBlock calls a built-in impulse() function to convert S-parameter S21 data into an impulse response. The user must provide physically realizable data to SBlock for a valid simulation. For example, if the S-parameter data represents a brick-wall filter, or a zero-delay filter, impulse() function will return an invalid impulse response without any warning.
4. Before you embed SBlock into a large system, specify N with different values as 64, 128, 256, 512 and so on to verify that SBlock accurately models your filter.
5. For information regarding timed linear component signals, refer to *Timed Linear Components* (timed).

SplitterRF



Description: RF signal splitter
Library: Timed, Linear
Class: TSDF_SplitterRF
Derived From: baseStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

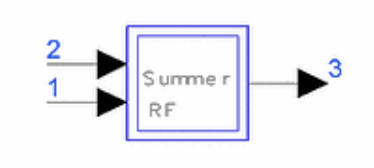
Pin Outputs

Pin	Name	Description	Signal Type
2	output1	output signal 1	timed
3	output2	output signal 2	timed

Notes/Equations

1. V3(t) = V2(t) = V1(t)
2. For information regarding timed linear component signals, refer to *Timed Linear Components* (timed).

SummerRF



Description: RF signal summer
Library: Timed, Linear
Class: TSDF_SummerRF
Derived From: baseStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
FcOut	output characterization frequency: min, center, max	max		enum	

Pin Inputs

Pin	Name	Description	Signal Type
1	input1	input signal 1	timed
2	input2	input signal 2	timed

Pin Outputs

Pin	Name	Description	Signal Type
3	output	output signal	timed

Notes/Equations

1. This component models a summer with two inputs.
   The FcOut parameter is used to select the output signal characterization frequency.
2. The following equations describe the algorithm used for this model.

   Note
   Warning messages will be displayed if the input signal simulation time step is too large to properly represent the output signal. A smaller value for the simulation time step is recommended when this happens.

   The signals $V_k(t)$ at pin k, where k = 1, 2, 3, are represented by their inphase and quadrature components about their carrier frequency:

   $$V_k(t) = Re\left\{v_k(t)e^{j2\pi f_{ck}t}\right\}, \quad v_k(t) = v_{Ik}(t) + jv_{Qk}(t)$$

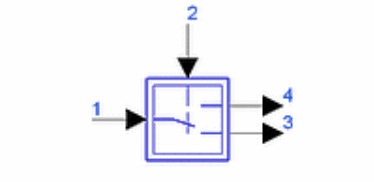
   The output signal $V_3(t)$ at pin 3 is given by the following equations:

$$f_{c3} = \begin{cases} 0 & \text{if } f_{c1}=0 \text{ or } f_{c2}=0 \\ min(f_{c1}, f_{c2}) & \text{if } f_{c1}>0 \text{ and } f_{c2}>0 \text{ and FcOut = min} \\ (f_{c1}+f_{c2})/2 & \text{if } f_{c1}>0 \text{ and } f_{c2}>0 \text{ and FcOut = center} \\ max(f_{c1}, f_{c2}) & \text{if } f_{c1}>0 \text{ and } f_{c2}>0 \text{ and FcOut = max} \end{cases}$$

$$V_3(t) = Re\left\{(v_1(t)e^{j2\pi(f_{c1}-f_{c3})t} + v_2(t)e^{j2\pi(f_{c2}-f_{c3})t})e^{j2\pi f_{c3}t}\right\}$$

3. Information regarding time domain signal differences between ADS Ptolemy simulations and Circuit Envelope and Transient simulations is given in the *Timed Synchronous Dataflow* (ptolemy) section in the *ADS Ptolemy Simulation* (ptolemy) documentation.
4. For information regarding timed linear component signals, refer to *Timed Linear Components* (timed).

SwitchSPDT



Description: RF single pole, double throw switch
Library: Timed, Linear
Class: TSDF_SwitchSPDT
Derived From: baseOmniSysStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
Loss1	loss in dB for on state insterion loss for output 1	0		real	[0, +200)
Isolation1	isolation in dB for off state insterion loss for output 1	-200		real	[0, +200)
Loss2	loss in dB for on state insterion loss for output 2	0		real	[0, +200)
Isolation2	isolation in dB for off state insterion loss for output 2	-200		real	[0, +200)
VThreshold	control voltage threshold	1	V	real	(0, ∞)
TOn1	on state transition time for output 1	0	sec	real	[0, ∞)
TOff1	off state transition time for output 1	0	sec	real	[0, ∞)
TOn2	on state transition time for output 2	0	sec	real	[0, ∞)
TOff2	off state transition time for output 2	0	sec	real	[0, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	input1	input signal 1	timed
2	input2	input signal 2	timed

Pin Outputs

Pin	Name	Description	Signal Type
3	output1	output signal 1	timed
4	output2	output signal 2	timed

Notes/Equations

1. SwitchSPDT can be used to model a non-ideal switch that passes its input signal through to one of two output pins. The non-idealities modeled are insertion losses, imperfect isolation, and non-zero delay switching times.
2. The following equations describe the algorithm used for this model.
   The input signal $V_1(t)$ at pin 1 is represented by its inphase and quadrature components about its carrier frequency:

$$V_1(t) = Re[v_1(t)e^{j2\pi f_{c1}t}], v_1(t) = v_{I1}(t) + jv_{Q1}(t)$$

Let V $_2$ *(t)* at pin 2 be the control signal.

Then, the output signals V $_3$ *(t)* and V $_4$ *(t)* at pins 3 and 4 are described by

$$V_3(t) = Re\left\{v_3(t)e^{j2\pi f_{c1}t}\right\}$$

$$V_4(t) = Re\left\{v_4(t)e^{j2\pi f_{c1}t}\right\}$$

where *v* $_3$ *(t)* and *v* $_4$ *(t)* are calculated as follows.

Case 1: *V* $_2$ *(t) > VTH* (this connects pin 4 to pin 1)

Let *T* $_s$ be the time at which V $_2$ *(t)* exceeds Vth. Then

$$v_3(t) = \begin{cases} 10^{-\frac{ISO2}{20}} v_1(t) & if\ t \ge T_s + Toff2 \\ \left(10^{-\frac{LOSS2}{20}} - 10^{-\frac{ISO2}{20}}\right)\left(1 - \frac{(t-T_s)}{Toff2}\right)v_1(t) + 10^{-\frac{ISO2}{20}} v_1(t) & otherwise \end{cases}$$

$$v_4(t) = \begin{cases} 10^{-\frac{LOSS1}{20}} v_1(t) & if\ t \ge T_s + Ton1 \\ \left(10^{-\frac{LOSS1}{20}} - 10^{-\frac{ISO1}{20}}\right)\frac{(t-T_s)}{Ton1} v_1(t) + 10^{-\frac{ISO1}{20}} v_1(t) & otherwise \end{cases}$$

Case 2: *V* $_2$ *(t) ≤ VTH* (this connects pin 3 to pin 1)

Let *T* $_s$ be the time at which *V* $_2$ *(t)* falls below *VTH*. Then

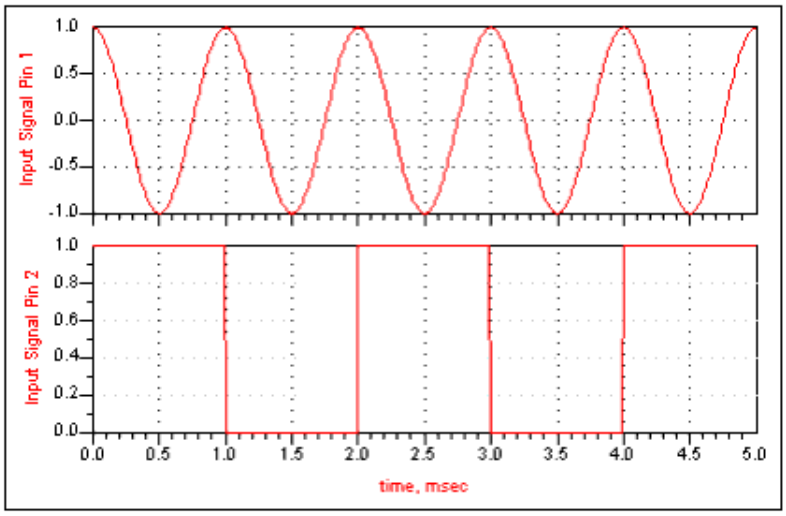
$$v_3(t) = \begin{cases} 10^{-\frac{LOSS2}{20}} v_1(t) & if\ t \ge T_s + Ton1 \\ \left(10^{-\frac{LOSS2}{20}} - 10^{-\frac{ISO2}{20}}\right)\frac{(t-T_s)}{Ton2} v_1(t) + 10^{-\frac{ISO2}{20}} v_1(t) & otherwise \end{cases}$$

$$v_4(t) = \begin{cases} 10^{-\frac{ISO1}{20}} v_1(t) & if\ t \ge T_s + Toff1 \\ \left(10^{-\frac{LOSS1}{20}} - 10^{-\frac{ISO1}{20}}\right)\left(1 - \frac{t-T_s}{Toff1}\right)v_1(t) + 10^{-\frac{ISO1}{20}} v_1(t) & otherwise \end{cases}$$

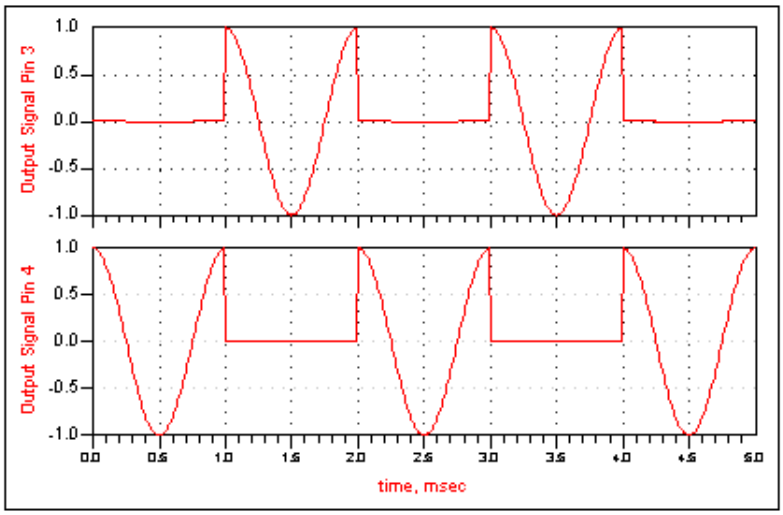
3. SwitchSPDT Component Inputs and SwitchSPDT Component Outputs show the performance of the SwitchSPDT component (when it is an ideal switch) using the following parameter values:
   Loss1=0, Isolation1=200,
   Loss2=0, Isolation2=200,
   VThreshold=0.50,
   TOn1=0, TOff1=0,
   TOn2=0, TOff2=0
   SwitchSPDT Component Outputs shows the outputs of the SwitchSPDT component (when the switch is not ideal) using the following parameter values:
   Loss1=3 dB, Isolation1=20 dB,

Loss2=3 dB, Isolation2=20 dB,
VThreshold=0.50,
TOn1=0.15 s, TOff1=0.15 s,
TOn2=0.15 s, TOff2=0.15 s

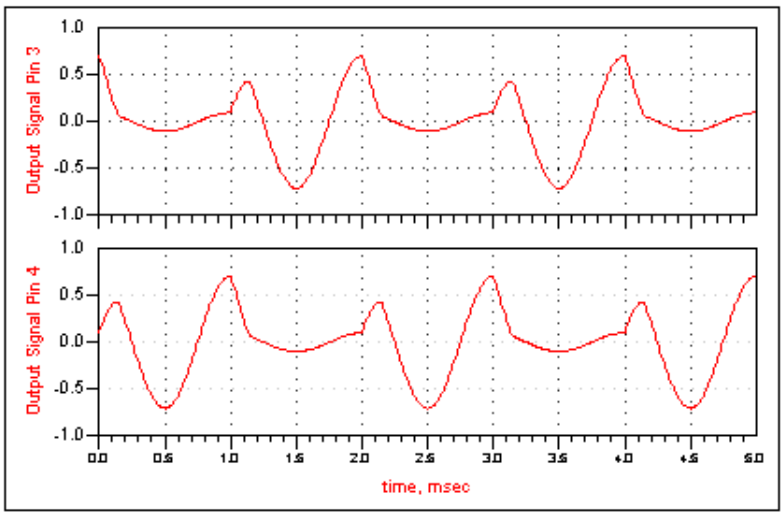
SwitchSPDT Component Inputs



SwitchSPDT Component Outputs

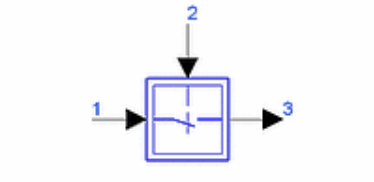


SwitchSPDT Component Outputs



4. For information regarding timed linear component signals, refer to *Timed Linear Components* (timed).

SwitchSPST



Description: RF single pole, single throw switch
Library: Timed, Linear
Class: TSDF_SwitchSPST
Derived From: baseOmniSysStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
Loss	loss in dB for on state insterion loss	0		real	[0, +200)
Isolation	Isolation in dB for off state insterion loss	-200		real	[0, +200)
VThreshold	control voltage threshold	1	V	real	(0, ∞)
TOn	on-state transition time for output	0	sec	real	[0, ∞)
TOff	off-state transition time for output	0	sec	real	[0, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	input1	input signal 1	timed
2	input2	input signal 2	timed

Pin Outputs

Pin	Name	Description	Signal Type
3	output	output signal	timed

Notes/Equations

1. This component can be used to model a non-ideal switch that passes its input signal through to its output pin. The non-idealities modeled are insertion losses, imperfect isolation, and non-zero delay switching times.
2. The following equations describe the algorithm used for this model.
   The input signal $V_1(t)$ at pin 1 is represented by its inphase and quadrature components about its carrier frequency:

$$V_1(t) = Re\left\{v_1(t)e^{j2\pi f_{c1}t}\right\} = Re\left\{(v_{I1}+jv_{Q1})e^{i2\pi f_{c1}t}\right\}$$

   Let $V_2(t)$ at pin 2 be the control signal.

   Then, the output signal $V_3(t)$ at pin 3 is described by the equation:

$$V_3(t) = Re\left\{v_3(t)e^{j2\pi f_{c1}t}\right\}$$

where $v_3(t)$ is computed as follows.

Case 1: $V_2(t) > VTH$

Let T_s be the time at which $V_2(t)$ exceeds VTH.

Then

$$v_3(t) = \begin{cases} 10^{-\frac{Loss}{20}} v_1(t) & if\ t \geq T_s + Ton \\ \left(10^{-\frac{Loss}{20}} - 10^{-\frac{Iso}{20}}\right)\frac{(t-T_s)}{Ton}v_1(t) + 10^{-\frac{Iso}{20}} v1(t) & otherwise \end{cases}$$

Case 2: $V_2(t) \leq VTH$

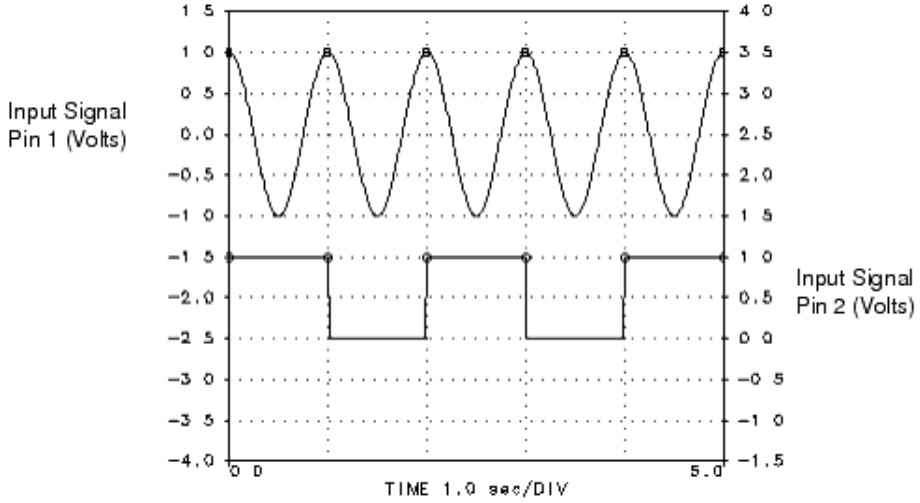
Let T_s be the time at which $V_2(t)$ falls below VTH.

Then

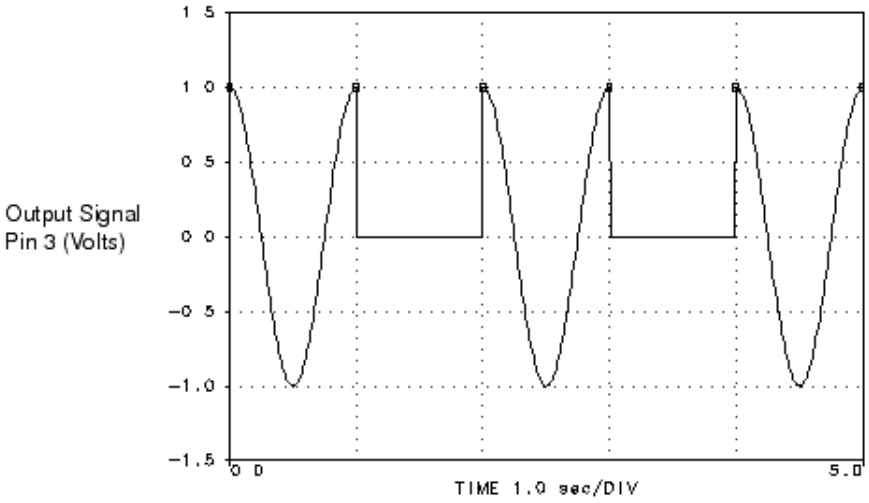
$$v_3(t) = \begin{cases} 10^{-\frac{ISO}{20}} v_1(t) & if\ t \geq T_s + Toff \\ \left(10^{-\frac{LOSS}{20}} - 10^{-\frac{ISO}{20}}\right)\left(1 - \frac{(t-T_s)}{Toff}\right)v_1(t) + 10^{-\frac{ISO}{20}} v1(t) & otherwise \end{cases}$$

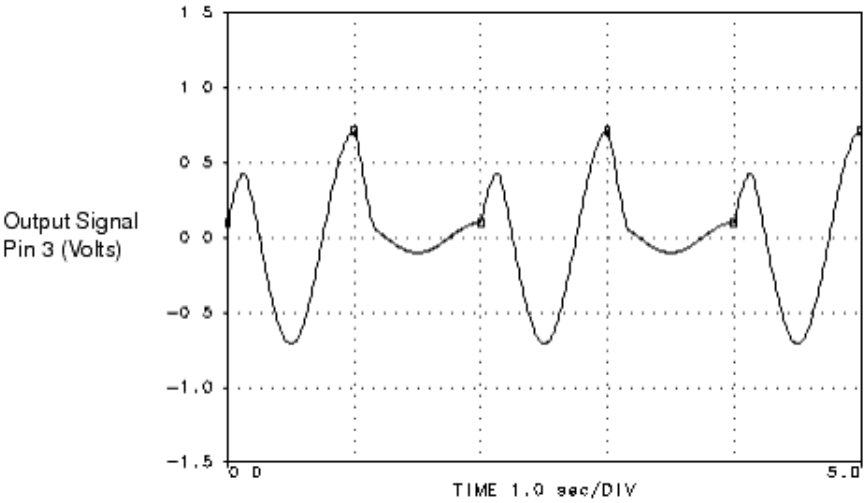
3. SwitchSPST Component Inputs and SwitchSPST Component Output show the performance of the SwitchSPST component, when it is an ideal switch, using the following parameter values:
Loss=0, Iso=200, Vth=0.50, Ton=0, Toff=0
SwitchSPST Component Output shows the output of the SwitchSPST component, when the switch is not ideal, using the parameter values:
Loss=3 dB, Iso=20 dB, Vth=0.50, Ton=0.15 s, Toff=0.15 s

SwitchSPST Component Inputs

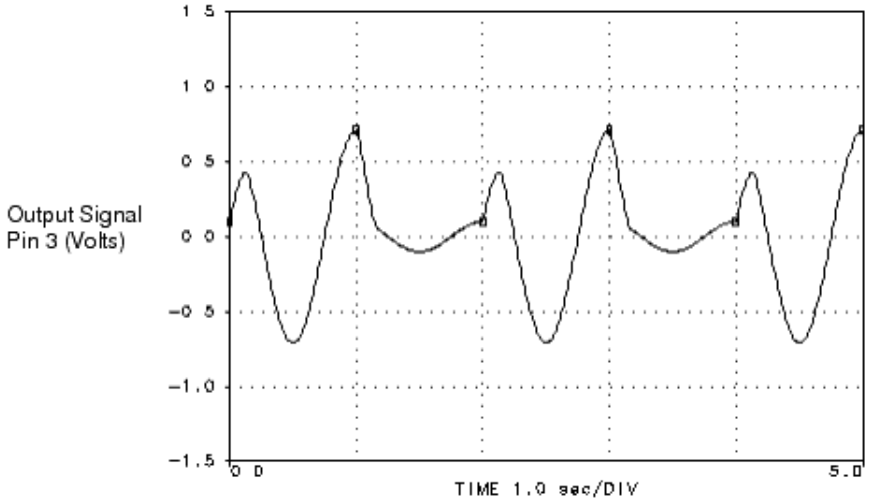


SwitchSPST Component Output



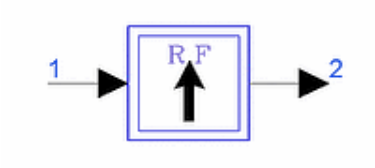


SwitchSPST Component Output



4. For information regarding timed linear component signals, refer to *Timed Linear Components* (timed).

USampleRF



Description: Up sampler for time domain signals
Library: Timed, Linear
Class: TSDF_USampleRF
Derived From: baseStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
Type	upsampling interpolation type: SampleAndHold, ZeroInsertion, PolyPhaseFilter, Linear	SampleAndHold		enum	
Ratio	upsampling ratio	5		int	[1, ∞)
InsertionPhase	upsampling insertion phase for the output non-zero sample when Type=ZeroInsertion	0		int	[0, Ratio - 1]
ExcessBW	excess bandwidth of raised cosine interpolation filter, used when Type=PolyPhaseFilter	0.5		real	[0, 1]

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
2	output	output signal	timed

Notes/Equations

1. This component upsamples an input timed signal to produce an output timed signal sampled with a time step equal to TStepOut = TStepIn / Ratio, where TStepIn is the input time step. For every one input sample consumed, Ratio output samples are produced (Ratio-1 new samples generated between each input signal pair).
2. When Type=SampleAndHold, the input sample is repeated Ratio times at the output. When Type=ZeroInsertion, the InsertionPhase parameter identifies which of the Ratio output samples will contain the one input sample. $v_2 \sim ((n \times Ratio+InsertionPhase) \times TStepOut) = v \sim_1 (n \times TStepIn)$, n is the input sample number, and all other values of $v_2(.)$ are zero.

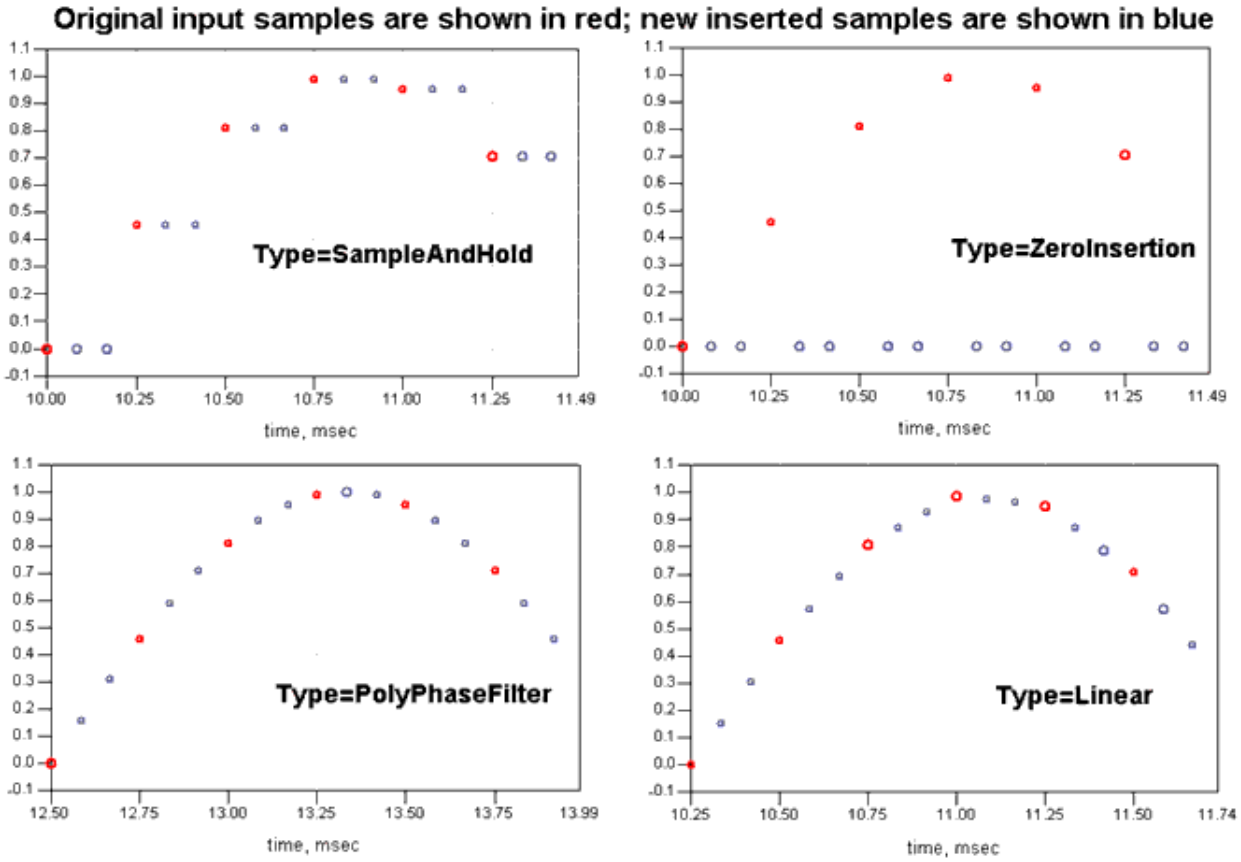
   When Type=PolyPhaseFilter, a raised-cosine filter with excess bandwidth equal to

ExcessBW (ExcessBW=0 gives the ideal lowpass filter) is used for interpolation. The corner frequency of the filter is set to FCorner=1/Ratio/TStepIn. The number of taps used for this filter is (1+20 × Ratio). Therefore, the output signal will be delayed with respect to the input signal by 10 × TStepIn.

When Type=Linear, linear interpolation is used to fill the values between consecutive input samples. The output signal will be delayed by TStepIn with respect to the input signal.

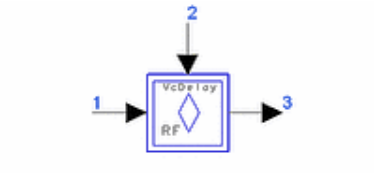
The figure below shows the USampleRF output for different Type values.

Upsampler outputs for different Type values: Type=ZeroInsertion plot created with InsertionPhase=0, Type=PolyPhaseFilter plot created with ExcessBW=0.5



3. To upsample a signal by a non-integer factor, a cascade of an upsampler and downsampler is needed. For example, to change the sampling rate of a signal from 40 MHz (TStep = 25 nsec) to 73 MHz (TStep = 13.69863 nsec), first pass it through the USampleRF component (Ratio = 73, Type=PolyPhaseFilter or Linear) and then through the DSampleRF component (Ratio=40). To improve simulation speed, make sure the two ratios are relatively prime; if not, divide them with their GCD (greatest common divisor). For example, if in the previous case the desired sampling rate was 72 MHz, Ratio of USampleRF can be set to 9 (=72/8) and Ratio of DSampleRF can be set to 5 (=40/8); GCD (72, 40) = 8.
4. For information regarding timed linear component signals, refer to *Timed Linear Components* (timed).

VcDelayRF



Description: Voltage controlled time delay
Library: Timed, Linear
Class: TSDF_VcDelayRF
Derived From: _DelayRF

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
InterpolationMethod	signal interpolation method: none, linear	none		enum	
IncludeCarrierPhaseShift	include RF carrier phase shift: No, Yes	Yes		enum	
MaxTimeDelay	maximum time delay available rounded to nearest TStep	0.0	sec	real	[TStep, ∞)†

† TStep is the simulation time step for the component input signals.

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed
2	control	control signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
3	output	output signal	timed

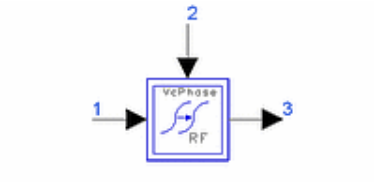
Notes/Equations

1. This component models a voltage-controlled time delay. The input signal is at pin 1, the control signal is at pin 2, and the delayed version of the pin 1 signal is at pin 3. The controlled delay is in addition to the built-in delay of one stimulation time step.
2. The time delay introduced is:
   Delay = (MaxTimeDelay - TStep)× control + TStep
   where
   TStep = one simulation time step
   control = signal at pin 2 for (0 ≤ V2 ≤ 1)
   = 0 for V2 < 0

= 1 for V2 > 1

3. The time delay operation is the same as described for the DelayRF component.
4. VcDelayRF delays the *input* signal at pin 1 only; it does not delay the *control* input signal at pin 2. Therefore, if the signal path from pin 2 to pin 3 is part of a feedback loop, a deadlock may be reported (if there is no delay in this loop). To solve this problem, a DelayRF component can be connected at the *control* input.
5. For information regarding timed linear component signals, refer to *Timed Linear Components* (timed).

VcPhaseShiftRF



Description: Voltage controlled RF signal phase shifter
Library: Timed, Linear
Class: TSDF_VcPhaseShiftRF
Derived From: baseStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
Sensitivity	phase shift sensitivity in degrees/Volt	0	deg	real	(-∞, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed
2	control	control signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
3	output	output signal	timed

Notes/Equations

1. This component models a voltage controlled phase shifter for an input RF signal. The input signal is at pin 1, the control signal is at pin 2, and the phase shifted version of the pin 1 signal is at pin 3.
2. The input signal must be an RF (complex envelope) signal represented by its inphase and quadrature components about its carrier frequency:

   $$V_1(t) = Re\left[v_1(t)e^{j2\pi f_{c1}t}\right], \mathrm{v}_1(t) = v_{I1}(t) + jv_{Q1}(t)$$

   The output signal at pin 3 is given by the equation:

   $$V_3(t) = Re\left[v_1(t)\, e^{j2\pi f_{c1}t + \phi}\right]$$

   where
   $\varphi = V_2(t) \times Sensitivity$

   If the input is a baseband signal ($f_{c1} = 0$) an error will be declared and simulation will stop.
   For information regarding timed linear component signals, refer to *Timed Linear*

3. *Components* (timed).

Timed Modem Components

The Timed Modem components library contains time domain modulators, demodulators, and carrier and clock recovery components.

Each timed modem component transmits, receives, or transforms time domain RF (complex envelope) signals. If a component receives a scalar signal other than a timed signal, the received signal is automatically converted to a timed (baseband) type of signal. Auto conversion from the complex scalar signal class to the timed (baseband) class is achieved by using the absolute value of the complex signal, sqrt(real ^{2} + imaginary ^{2}). These components do not accept any matrix class of signal.

Some components will only accept or produce a baseband or RF (complex envelope) timed signal. If a component requires an RF (complex envelope) timed signal, then its receipt of a baseband timed signal will be declared an error and the simulation will stop. If a baseband timed signal is the required input, then any received RF (complex envelope) timed signals is first transformed into its baseband equivalent form before use by the component.

An RF (complex envelope) timed signal is converted to its equivalent baseband form as follows:

$$V_{bb}(t) = Re\left\{(v_{RF}(t))e^{j2\pi f_c t}\right\} = Re\left\{(v_I(t) + jv_Q(t))e^{j2\pi f_c t}\right\}$$

where
$V_{bb}(t)$ is the total representation for the RF signal (also called the baseband representation)
$v_{RF}(t)$ is the RF signal complex envelope at characterization frequency fc (also called the equivalent complex baseband envelope representation for the RF signal)
$v_I(t)$ is the RF timed signal in-phase envelope
$v_Q(t)$ is the RF timed signal quadrature-phase envelope
f_c is the RF signal characterization frequency

For this equivalence to be valid, the simulation time step must be less than the inverse of the characterization frequency and the RF signal information content has an information bandwidth less than the RF characterization frequency.

The modulator components input baseband messaging signals and output the analog or digital modulation of RF signals. The output RF signal is represented by its complex envelope and its carrier frequency. A basic assumption that is made when using such a representation is that the carrier frequency is greater than the bandwidth of the message signal. The user must ensure that this condition is met when using these components.

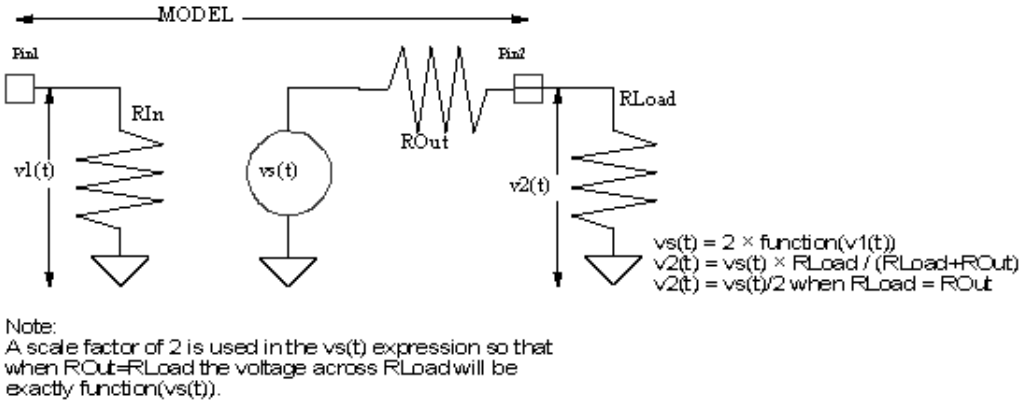
The carrier and clock recovery components input modulated RF signals and act to suppress the messaging content on the signal while preserving the RF carrier frequency and phase. In general, the spectral purity of the recovered carriers increases as the filtering bandwidth decreases-at the cost of increased initial transient times and delay through the component. Therefore, application of these components depends on the acceptance of such delays.

The demodulator components accept input RF signals and output the message signal content of the modulated RF signals as baseband signals. A basic assumption is that the input RF carrier frequency is greater than the bandwidth of the message signals. The user must ensure that this condition is met when using these components.

All modem components have user-specified input resistance (RIn) and output resistance (ROut). Input resistance is for a shunt resistor at each component input pin; output resistance is for a series resistor at each component output pin.

Each component internal output signal vs(t) has a value equal to twice the output pin signal v2(t) when the output is connected to a matched resistor load. The circuit model in 2-Port Circuit Model for a 2-port (1 input, 1 output) timed modem component demonstrates this relationship.

2-Port Circuit Model



Note:
A scale factor of 2 is used in the vs(t) expression so that when ROut=RLoad the voltage across RLoad will be exactly function(vs(t)).

The output pin signal v2(t) at the output series resistance is dependent on the value of the load resistance connected to it. When the load resistor RLoad is equal to the model output resistor ROut the value of v2(t) is equal to vs(t)/2; otherwise, based on the voltage divider action, v2(t) is:
$v\,2\,(t) = vs(t) \times RLoad/(RLoad + ROut\ )$

The input and output resistors must be greater than zero.

The input and output resistors contribute additive thermal noise power (kTB) to the output signal when the specified resistance temperature (RTemp) is greater than absolute zero (−273.15 °C) where
k = Boltzmann's constant
T = temperature in Kelvin
B = simulation frequency bandwidth:
1/2/tstep if signal is a timed baseband signal;
1/tstep if signal is a timed complex envelope signal

When RTemp > -273.15, the noise contributed from each resistor instance is an independent noise process. This noise is dependent on the value of DefaultSeed in the DF (data flow) controller. When DefaultSeed = 0, the noise generated for each simulation is different. When DefaultSeed > 0, the noise generated for each simulation, though

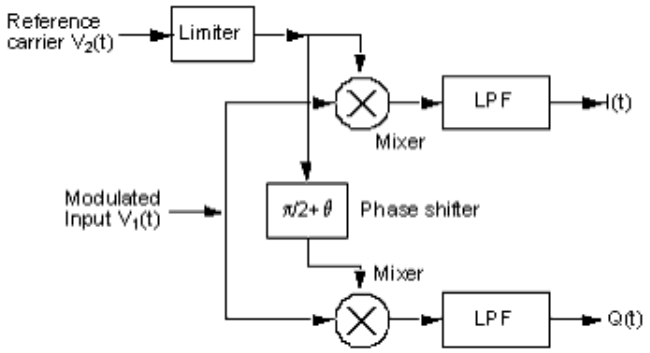
random, has the same initial seed starting condition and thus results in reproducible noise for each simulation.

Note
Information regarding time domain signal differences between ADS Ptolemy simulations and Circuit Envelope and Transient simulations is given in the *Timed Synchronous Dataflow* (ptolemy) section in the *ADS Ptolemy Simulation* (ptolemy) documentation.

Demodulation Process

For all demodulator components, the inphase and quadrature components of the modulated signal are extracted as the first step in the demodulation process as shown in Demodulation Process.

Demodulation Process



The circuit shown in Demodulation Process requires a reference carrier whose frequency and phase is synchronized with the transmitted signal. The modulated signal is mixed with the reference carrier and filtered to obtain the inphase component I(t). The reference carrier is phase shifted by 90 degrees (ideally), then mixed with the modulated signal and filtered to produce the quadrature component Q(t) (in demodulators QAM_Demod and QAM_DemodExtOsc a non-ideal phase shifter can be specified by setting the angle θ to a non-zero value through the parameter PhaseImbalance). Note that the reference carrier is passed through a limiter that removes any amplitude variations in the signal. Therefore, the demodulated outputs are not sensitive to the amplitude of the reference signal.

Demodulator Models

Two different models are provided for many of the demodulators that allow the reference carrier to be specified in different ways.

The first model applies to the components with an internal oscillator, including AM_Demod, FM_Demod, PM_Demod, QAM_Demod and DetectorRF. In these components the reference signal $V_2(t)$ is generated internally by an oscillator and the user can specify its frequency and phase by the parameters RefFreq and Phase, respectively. The oscillator generates an ideal sinusoid whose frequency can be set equal to the carrier frequency of the input signal; or, the effects of a frequency offset can be observed by setting it to a different frequency. Specifying the correct phase of the local oscillator is, in general, a more difficult task because the user must account for all phase shifts that the modulated signal undergoes from the transmitter to the receiver. The phase angle of the local oscillator must then be offset by this same amount to correctly demodulate the signal.

Frequency synchronization is automated by setting RefFreq=−1. The program then automatically determines the frequency of the modulated signal and adjusts the local oscillator accordingly, but its phase is set equal to 0.

The following equations describe how the reference signal $V_2(t)$ is specified for different cases. Let the modulated signal be represented in terms of its inphase and quadrature phase components.

$$V_1(t) = Re\left\{ v_1(t) e^{j2\pi f_{c1} t} \right\}, \; v_1(t) = v_{I1}(t) + j v_{Q1}(t)$$

Then

$$V_2(t) = Re\left\{ v_2(t) e^{j2\pi f_{c2} t} \right\}, \; v_2(t) = v_{I2}(t) + j v_{Q2}(t)$$

where

$$f_{c2} = \begin{cases} f_{c1} & \text{when RefFreq} = -1 \\ Fc & \text{when RefFreq} > 0 \end{cases}$$

$$v_2(t) = e^{j\varphi}$$

Here j is the phase angle specified by the parameter Phase.

Caveat: The previous equations show that when RefFreq=−1, frequency synchronization is achieved by simply setting $f_{c2} = f_{c1}$. However, it is not always true that f_{c1} equals the carrier frequency of the RF signal. It is possible to characterize the modulated signal about a frequency different from its carrier frequency by using the component FcChange. In this case the frequency synchronization algorithm will not work. It is also clear that if $f_{c1} = 0$, then frequency synchronization cannot be achieved by setting the parameter RefFreq=−1. Instead, the parameter RefFreq must be set explicitly to the modulation frequency of the signal.

The second model applies to components that use an external oscillator, including AM_DemodExtOsc, FM_DemodExtOsc, PM_DemodExtOsc, and QAM_DemodExtOsc. For these components the reference carrier must be provided from an external source to the demodulator-for which a second input pin is provided. The reference carrier must either be extracted from the received signal using a carrier recovery circuit or can be generated using the OSC component. Performance of carrier recovery circuits, effects of phase noise on the demodulation process, and so on, can then be studied.

The external reference carrier must be represented by its complex envelope with respect to a carrier frequency $f_{c2} > 0$. This is because the phase shifter shown in Demodulation Process is implemented by rotating the inphase and quadrature phase components of the reference carrier by the required angle, and this requires that these components be calculated with respect to a nonzero carrier frequency.

Demodulation Algorithms

The algorithms used to demodulate the signal can be broken into two cases: case 1 is for $f_{c1} > 0$; case 2 is for $f_{c1} = 0$.

In case 1 the modulated signal $V_1(t)$ is represented in terms of its inphase and quadrature components with respect to a carrier frequency > 0, that is,

$$V_1(t) = Re\left\{v_1(t)e^{j2\pi f_{c1}t}\right\}, \qquad f_{c1} > 0$$

where

$$v_1(t) = v_{I1}(t) + jv_{Q1}(t)$$

As noted earlier, the reference carrier $V_2(t)$ must be represented in terms of its inphase and quadrature components about a carrier frequency > 0. The outputs are then determined:

$$I(t) = Re\left\{v_1(t)\frac{v_2^{\dagger}(t)}{|v_2(t)|}e^{j2\pi(f_{c1}-f_{c2})t}\right\}$$

$$Q(t) = Re\left\{v_1(t)\frac{v_2^{\dagger}(t)}{|v_2(t)|}e^{j2\pi(f_{c1}-f_{c2})t-\left(\frac{\pi}{2}+\theta\right)}\right\}$$

Here $v_2^{\dagger}(t)$ is the complex conjugate of $v_2(t)$ and θ can be set to a nonzero value only in the QAM_Demod and QAM_DemodOsc components by the parameter PhaseImbalance.

In case 2, when $f_{c1} = 0$, an explicit filtering operation is performed to obtain the inphase and quadrature components:

$$I(t) = \left(V_1(t)Re\left\{\frac{v_2^{\dagger}(t)}{|v_2(t)|}e^{j2\pi f_{c2}t}\right\}\right) \otimes h_{lpf}(t)$$

$$Q(t) = \left(V_1(t)Re\left\{\frac{v_2^{\dagger}(t)}{|v_2(t)|}e^{j\left(2\pi f_{c2}t+\frac{\pi}{2}\right)}\right\}\right) \otimes h_{lpf}(t)$$

Here $h(t)$ is the impulse response of a lowpass filter. The lowpass filter is the same as

lpf

the timed raised-cosine filter LPF_RaisedCosineTimed with parameters:
CornerFreq = f_{c2} / 2

ExcessBW = 0.5
Type = impulse
SquareRoot = No
Delay = 2 / f_{c2}

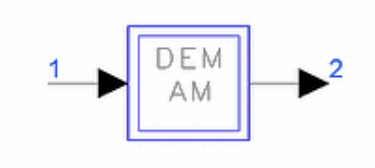
WindowType = Hamming

The output will therefore be delayed and will have some initial start-up transients associated with the filtering operation.

Components

- *AM Demod* (timed)
- *AM DemodExtOsc* (timed)
- *AM Mod* (timed)
- *DBPSK Demod* (timed)
- *DBPSK Mod* (timed)
- *DetectorRF* (timed)
- *DQPSK Demod* (timed)
- *DQPSK Mod* (timed)
- *DQPSK Pi4Demod* (timed)
- *DQPSK Pi4DemodSync* (timed)
- *DQPSK Pi4Mod* (timed)
- *DQPSK Pi4Recovery* (timed)
- *FM Demod* (timed)
- *FM DemodExtOsc* (timed)
- *FM Mod* (timed)
- *GMSK Demod* (timed)
- *GMSK Mod* (timed)
- *GMSK Recovery* (timed)
- *MSK Demod* (timed)
- *MSK Mod* (timed)
- *MSK Recovery* (timed)
- *PM Demod* (timed)
- *PM DemodExtOsc* (timed)
- *PM Mod* (timed)
- *QAM Demod* (timed)
- *QAM DemodExtOsc* (timed)
- *QAM Mod* (timed)
- *QAM ModExtOsc* (timed)
- *QPSK Demod* (timed)
- *QPSK Mod* (timed)
- *QPSK ModExtOsc* (timed)
- *QPSK Recovery* (timed)

AM_Demod



Description: Amplitude demodulator with internal oscillator
Library: Timed, Modem
Class: TSDF_AM_Demod
Derived From: baseOmniSysStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
RefFreq	internal reference frequency	1000000	Hz	real	{-1} or (0, ∞) †
Sensitivity	voltage output sensitivity, Vout/Vin	1		real	(-∞, ∞)
Phase	reference phase, in degrees	0	deg	real	(-∞, ∞)

† Refer to Note 2 for the special meaning of a -1 value.

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

Pin Outputs

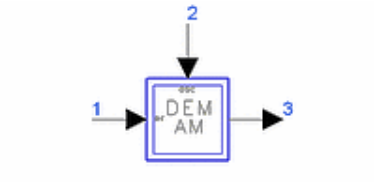
Pin	Name	Description	Signal Type
2	output	output signal	timed

Notes/Equations

1. An internal oscillator generates the reference carrier signal used to demodulate the RF signal. The inphase component I(t) of the modulated signal is extracted according to the procedure described in *Timed Modem Components* (timed). Then $V_3(t) = S\ I(t)$.
2. When RefFreq = −1, then internal oscillator frequency synchronization to the input signal is performed as described in *Timed Modem Components* (timed). This synchronization is allowed only when the input signal is an RF (complex envelope) timed signal.
3. When the input signal is a baseband signal, input signal transformation to an RF (complex envelope) timed signal is performed as described in *Demodulation Algorithms* (timed), case 2, before the demodulation process is performed.
4. AM_Demod is not a non-synchronous RF envelope detector; such a detector is

defined by the *DetectorRF* (timed) component.

AM_DemodExtOsc



Description: Amplitude demodulator with external reference oscillator
Library: Timed, Modem
Class: TSDF_AM_DemodExtOsc
Derived From: baseOmniSysStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
Sensitivity	voltage output sensitivity, Vout/Vin	1		real	(-∞, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed
2	osc	RF oscillator signal	timed

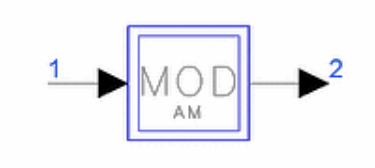
Pin Outputs

Pin	Name	Description	Signal Type
3	output	output signal	timed

Notes/Equations

1. An external reference carrier signal is required to demodulate the RF signal. The inphase components I(t) of the modulated signal is extracted according to the procedure described in *Timed Modem Components* (timed). Then $V_3(t)=S\ I(t)$.
2. When the input signal is a baseband signal, input signal transformation to an RF (complex envelope) timed signal is performed as described in *Demodulation Algorithms* (timed), case 2, before the demodulation process is performed.

AM_Mod



Description: Amplitude modulator with internal oscillator
Library: Timed, Modem
Class: TSDF_AM_Mod
Derived From: baseModulator

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
FCarrier	carrier frequency	1000000	Hz	real	(0, ∞)
Power	unmodulated carrier power	0.01	W	real	[0, ∞)
VRef	modulated voltage reference level	1	V	real	(0, ∞)
Type	modulation type: Conventional Am, DSB SC Am	Conventional Am		enum	

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
2	output	output signal	timed

Notes/Equations

1. This component is a modulator, where the modulation frequency and output power can be specified by the user.
2. The output of a Conventional AM modulator can be represented as:

   $$V_2(t) = A\left(1 + \frac{V_1(t)}{VRef}\right)\cos(\omega_c t)$$

   The output of a DSB SC AM modulator can be represented as:

   $$V_2(t) = A\left(\frac{V_1(t)}{VRef}\right)\cos(\omega_c t)$$

   where A is the unmodulated carrier amplitude and ωc is the carrier frequency. In the program the signal $V_2(t)$ is represented in terms of its inphase and quadrature components:

$$V_2(t) = Re\left\{(v_{I2}(t) + jv_{Q2}(t))e^{j\omega_c t}\right\}$$

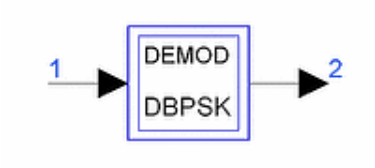
$$v_{I2}(t) = A\left(1 + \frac{V_1(t)}{V_{Ref}}\right),\ v_{Q2}(t) = 0 \text{ for Conventional AM}$$

$$v_{I2}(t) = A\left(\frac{V_1(t)}{V_{Ref}}\right),\ v_{Q2}(t) = 0 \text{ for DSBSc AM}$$

$$A = \sqrt{(2)(Rout)10^{\frac{PdBm - 30}{10}}},\ \omega_c = 2\pi f_c$$

where
PdBm = power in dBm units
f_c = FCarrier

DBPSK_Demod



Description: Non-coherent differential BPSK demodulator
Library: Timed, Modem
Class: TSDFDBPSK_Demod
Derived From: baseSubcircuit

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
RefFreq	internal reference frequency	1000000	Hz	real	(0, ∞)
SymbolTime	input symbol time interval	0.001	sec	real	[TStep, ∞)†
ExcessBw	raised cosine filter excess bandwidth	0.5		real	[newpro:0, 1]

† TStep is the simulation time step for the component input signal.

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
2	output	output signal	timed

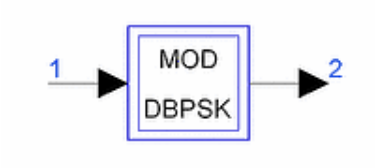
Notes/Equations

1. This component accepts an RF DQPSK signal at its input and non-coherently demodulates it. A local oscillator is not used in this demodulation process; this is possible because of the differential encoding performed at the transmitter. For each input sample consumed one output sample is produced.
2. The following points should be noted about the demodulator.
   - The input RF signal is filtered with a bandpass raised-cosine filter that is centered at the frequency specified by the parameter RefFreq. The filter has a bandwidth of 1 / (2SymbolTime) Hz and its rolloff factor is set by the parameter ExcessBw.
   - One-half of the raised-cosine filtering is provided in the receiver (by using a square root raised-cosine filter); the remaining one-half of the filtering should

be performed at the transmitter.

- The impulse response of the raised-cosine filter has a delay of 4SymbolTime seconds. This delay should be increased to 8SymbolTime or greater for better accuracy, especially when making measurements such as EVM. The DBPSK_Mod component has a similar delay of 4SymbolTime seconds. Therefore, the total delay introduced by the modulator and demodulator is 8SymbolTime seconds.
- The first decoded symbol may be in error because the initial state of the differential encoder is not known by the receiver.

DBPSK_Mod



Description: Differential BPSK modulator with internal oscillator
Library: Timed, Modem
Class: TSDFDBPSK_Mod
Derived From: baseSubcircuit

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
FCarrier	carrier frequency	1000000	Hz	real	(0, ∞)
SymbolTime	input symbol time interval	0.001	sec	real	[TStep, ∞)†
Power	modulator output power	0.01	W	real	[0, ∞)
ExcessBw	raised cosine filter excess bandwidth	0.5		real	[newpro:0, 1]

† TStep is the simulation time step for the component input signal.

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

Pin Outputs

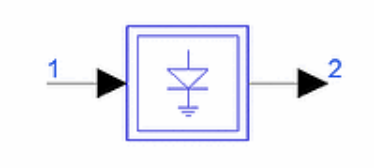
Pin	Name	Description	Signal Type
2	output	output signal	timed

Notes/Equations

1. This component accepts a baseband NRZ data stream at its input and generates a DBPSK RF signal at its output. The input NRZ data stream is first differentially encoded, then filtered with a raised-cosine filter and finally upconverted to the carrier frequency. For each input sample consumed one output sample is produced.
2. The following points should be noted about this modulator:
   - The symbol period of the input data is specified by the parameter SymbolTime; the rolloff factor of the filter is set by the parameter ExcessBw; and, the carrier frequency and power of the output signal are set by the parameters FCarrier and Power, respectively. In order to get the correct power at the output of the modulator, the input NRZ waveform must have an amplitude of 1V.

- One-half of the raised-cosine filtering is provided in the transmitter (by using a square root raised-cosine filter); the remaining half of the filtering should be performed at the receiver.
- The filter has an f/sin(f) shaped equalization to account for the finite width of the input NRZ data.
- The raised-cosine filter has a corner frequency of 1 / 2SymbolTime Hz.
- The impulse response of the filter has a delay of 4SymbolTime seconds. This delay should be increased to 8SymbolTime or greater for better accuracy, especially when making measurements such as EVM.

DetectorRF



Description: RF envelope detector with internal oscillator
Library: Timed, Modem
Class: TSDF_DetectorRF
Derived From: baseOmniSysStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
RefFreq	carrier frequency	1000000	Hz	real	{-1} or (0, ∞)†
Sensitivity	voltage output sensitivity, Vout/Vin	1		real	(-∞, ∞)
Type	detector type: Rectified envelope, Inphase envelope, Quadrature envelope	Rectified envelope		enum	

† Refer to Note 2 for the special meaning of a -1 value.

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
2	output	output signal	timed

Notes/Equations

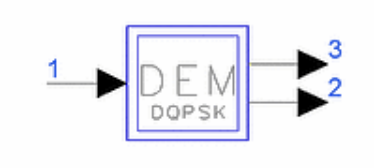
1. This envelope detector can be used to either recover the envelope of an RF signal or its rectified inphase or quadrature components. The output of the detector is always a baseband signal. The effect of a frequency mismatch can be observed by setting RefFreq to a value different from that of the input signal carrier frequency.
2. When RefFreq = −1, then internal oscillator frequency synchronization to the input signal is performed as described in the *Introduction".* (timed) This synchronization is only allowed when the input signal is an RF (complex envelope) timed signal. The envelope of the input signal will be the output.
3. When the input signal is a baseband signal, input signal transformation to an RF (complex envelope) timed signal is performed as described in *Demodulation*

Algorithms (timed), case 2, before the demodulation process is performed.

4. The inphase and quadrature components I(t) and Q(t) of the modulated signal are extracted according to the procedure described in *Timed Modem Components* (timed). The output signal *V* 2 *(t)* is then calculated as:

$$V_2(t) = \begin{cases} S\sqrt{I^2(t) + Q^2(t)} & \text{when Type = Rectified envelope} \\ S|I(t)| & \text{when Type = Inphase envelope} \\ S|Q(t)| & \text{when Type = Quadrature envelope} \end{cases}$$

DQPSK_Demod



Description: Non-coherent differential QPSK demodulator
Library: Timed, Modem
Class: TSDFDQPSK_Demod
Derived From: baseSubcircuit

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
RefFreq	internal reference frequency	1000000	Hz	real	(0, ∞)
SymbolTime	input symbol time interval	0.001	sec	real	[TStep, ∞)†
ExcessBw	raised cosine filter excess bandwidth	0.5		real	[newpro:0, 1]

† TStep is the simulation time step for the component input signal.

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

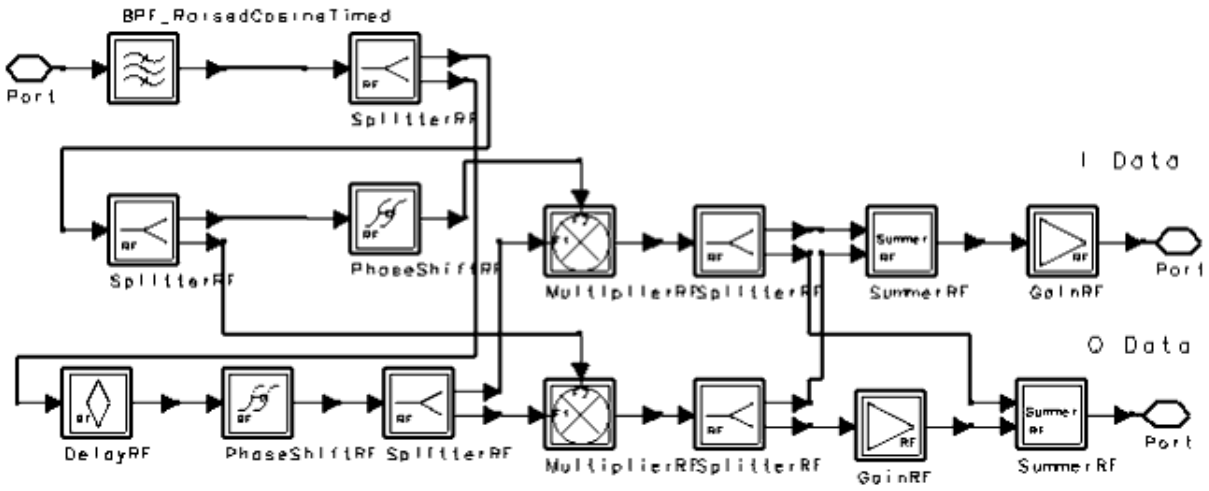
Pin Outputs

Pin	Name	Description	Signal Type
2	I_out	I output	timed
3	Q_out	Q output	timed

Notes/Equations

1. This component is composed of other components. DQPSK Demodulator Network shows the implementation of the network representing the demodulator. A local oscillator is not used in this demodulation scheme; this is possible because of the differential encoding performed at the transmitter.

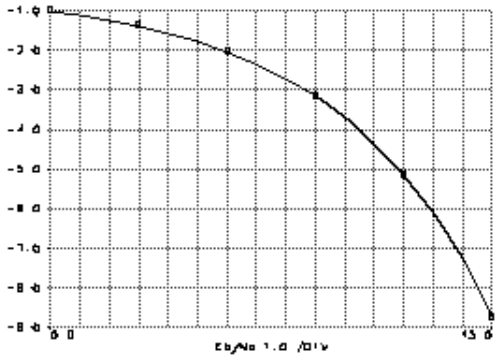
DQPSK Demodulator Network



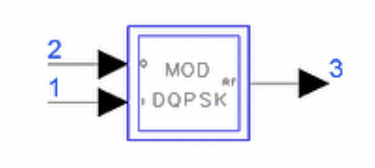
The following points should be noted about the demodulator.

- The RF signal is filtered with a bandpass raised-cosine filter that is centered at the frequency specified by the parameter RefFreq. The filter has a bandwidth of (1/(2SymbolTime)) Hz and its rolloff factor is set by the parameter ExcessBw.
- One-half of the raised-cosine filtering is provided in the receiver (by using the square root of the cosine filter); the remaining one-half of the filtering should be performed at the transmitter.
- The impulse response of the raised-cosine filter has a delay of 4SymbolTime seconds. This delay should be increased to 8SymbolTime or greater for better accuracy, especially when making measurements such as EVM. The DQPSK_Mod component has a similar delay of 4SymbolTime seconds and an additional delay of 1SymbolTime seconds due to the differential encoding. Therefore, the total delay introduced by the modulator and demodulator is 9SymbolTime seconds.
- The first I and Q decoded data bits may be in error because the initial state of the differential encoder is not known by the receiver.
- The required input RF power level in order to obtain ±1V at the ideal sampling instances at the output of this demodulator is 13 dBm.
- The BER performance of the DQPSK_Mod/DQPSK_Demod pair for a distortionless additive white Gaussian channel is shown in BER Performance of DQPSK_Demod.

BER Performance of DQPSK_Demod



DQPSK_Mod



Description: Differential QPSK modulator with internal oscillator
Library: Timed, Modem
Class: TSDFDQPSK_Mod
Derived From: baseSubcircuit

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
FCarrier	carrier frequency	1000000	Hz	real	(0, ∞)
SymbolTime	input symbol time interval	0.001	sec	real	[TStep, ∞)†
Power	modulator output power	0.01	W	real	[0, ∞)
ExcessBw	raised cosine filter excess bandwidth	0.5		real	[newpro:0, 1]

† TStep is the simulation time step for the component input signals.

Pin Inputs

Pin	Name	Description	Signal Type
1	I_in	I input	timed
2	Q_in	Q input	timed

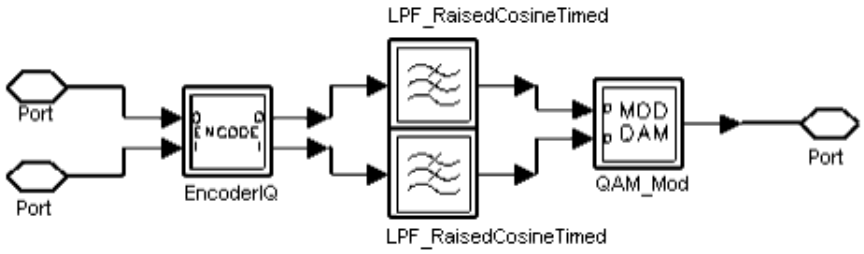
Pin Outputs

Pin	Name	Description	Signal Type
3	output	output signal	timed

Notes/Equations

1. This component is a hierarchical model composed of other components. DQPSK Modulator Schematic shows the implementation of the network representing the component.
   The two inputs to the modulator are the I and Q data streams in NRZ format. The NRZ data symbols are differentially encoded for the DQPSK format, filtered with raised-cosine filters and then modulated to the carrier frequency with a QAM modulator.

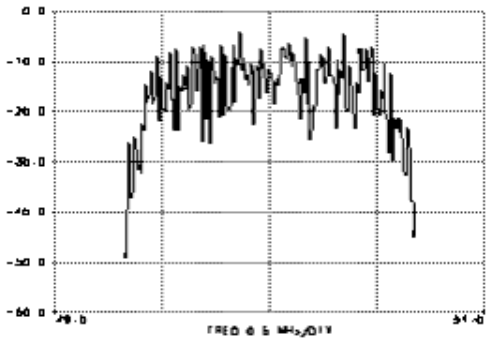
DQPSK Modulator Schematic



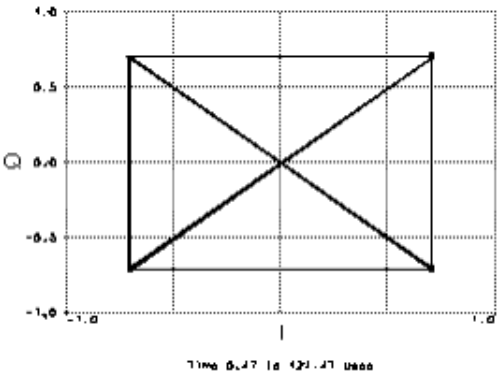
The following points should be noted about the transmitter:

- The bit time of the input data is specified by parameter SymbolTime; the roll off factor of the filters is set by parameter ExcessBw; and, the carrier frequency and power of the output signal are set by parameters FCarrier and Power, respectively.
- One-half of the raised-cosine filtering is provided in the transmitter (by using the square root of the cosine filters); the remaining half of the filtering should be performed at the receiver.
- The filters have an (f/sin(f)) shaped equalization to account for the finite width of the input NRZ data.
- The raised-cosine filters have a corner frequency of 1/2SymbolTime Hz.
- The impulse response of the filters has a delay of 4SymbolTime seconds. This delay should be increased to 8SymbolTime or greater for better accuracy, especially when making measurements such as EVM.
  The spectrum of the modulated signal is shown in Output SPECTRUM (Power (dBm) versus Freq) of DQPSK Modulator for the following values of the parameters: SymbolTime=1 µsec., FCarrier=50 MHz, Power=10 dBm, ExcessBw=0.35.
  The IQ constellation diagram of the modulated signal, after filtering with a bandpass square root raised-cosine filter, is shown in DQPSK (I versus Q) Constellation Diagram.

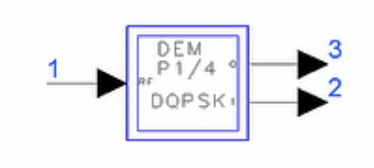
Output SPECTRUM (Power (dBm) versus Freq) of DQPSK Modulator



DQPSK (I versus Q) Constellation Diagram


Q
I

DQPSK_Pi4Demod



Description: Non-coherent pi/4-DQPSK demodulator
Library: Timed, Modem
Class: TSDFDQPSK_Pi4Demod
Derived From: baseSubcircuit

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
RefFreq	internal reference frequency	1000000	Hz	real	(0, ∞)
SymbolTime	input symbol time interval	0.001	sec	real	[TStep, ∞)†
ExcessBw	raised cosine filter excess bandwidth	0.5		real	[newpro:0, 1]

† TStep is the simulation time step for the component input signal.

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

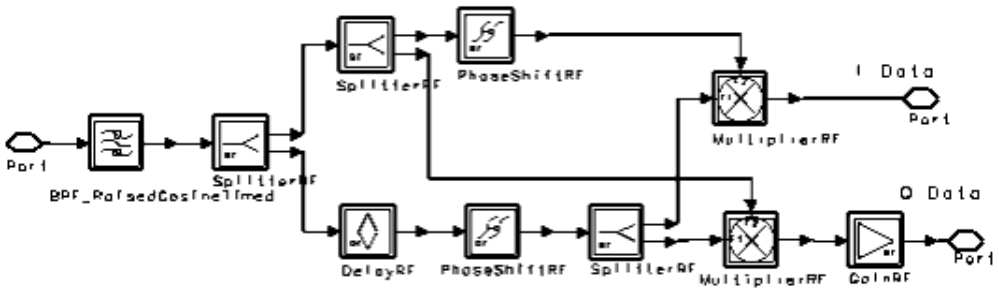
Pin Outputs

Pin	Name	Description	Signal Type
2	I_out	I output	timed
3	Q_out	Q output	timed

Notes/Equations

1. This component is a hierarchical model composed of other components. PI4DQPSK Demodulator Network shows the implementation of the network representing the demodulator. A local oscillator is not used in this demodulation scheme; this is possible because of the differential encoding performed at the transmitter.

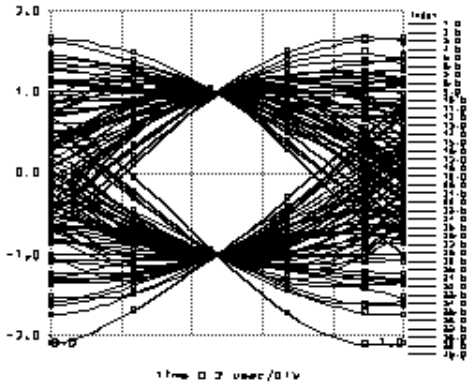
PI4DQPSK Demodulator Network



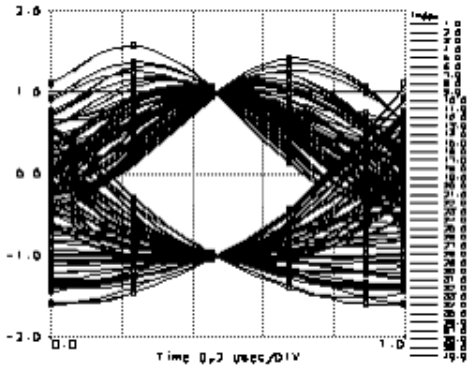
The following points should be noted about the demodulator.

- The RF signal is filtered with a bandpass raised-cosine filter that is centered at the frequency specified by the parameter RefFreq. The filter has a bandwidth of 1/2SymbolTime Hz and its rolloff factor is set by the parameter ExcessBw.
- One-half of the raised-cosine filtering is provided in the receiver (by using the square root of the cosine filter); the remaining half of the filtering should be performed at the transmitter.
- The impulse response of the raised-cosine filter has a delay of 4SymbolTime seconds. The modulator component DQPSK_Pi4Mod has a similar delay of 4SymbolTime seconds and an additional delay of 1SymbolTime seconds due to the differential encoding. Therefore, the total delay introduced by the modulator and demodulator is 9SymbolTime seconds.
- The very first I and Q decoded data bits may be in error because the initial state of the differential encoder is not known by the receiver.
- The required input RF power level in order to obtain ±1V at the ideal sampling instances at the output of this demodulator is 14.5 dBm.
- The eye pattern for the demodulated I output is shown in Eye Pattern for Demodulated I Output; the eye pattern for the demodulated Q output is shown in Eye Pattern for Demodulated Q Output; and, the I-Q diagram for the demodulated output is shown in I-Q Diagram for Demodulated Output.
- The BER performance of the DQPSK_Pi4Mod/DQPSK_Pi4Demod pair for a distortionless additive white Gaussian channel is shown in BER Performance of DQPSK_Pi4Mod/DQPSK_Pi4Demod Pair.

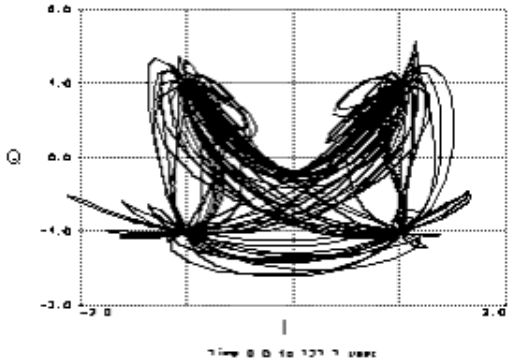
Eye Pattern for Demodulated I Output



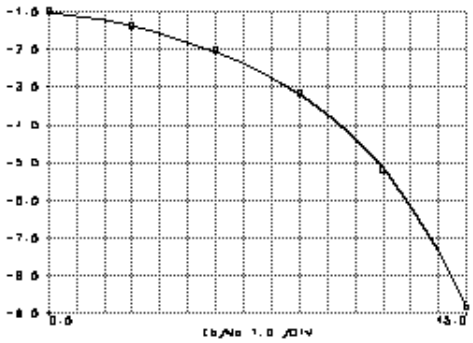
Eye Pattern for Demodulated Q Output



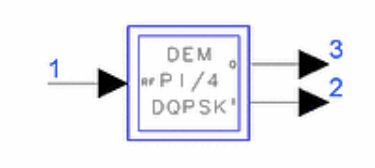
I-Q Diagram for Demodulated Output



BER Performance of DQPSK_Pi4Mod/DQPSK_Pi4Demod Pair



DQPSK_Pi4DemodSync



Description: Synchronous pi/4-DQPSK demodulator with internal oscillator
Library: Timed, Modem
Class: TSDFDQPSK_Pi4DemodSync
Derived From: baseSubcircuit

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
RefFreq	internal reference frequency	1000000	Hz	real	
SymbolTime	input symbol time interval	0.001	sec	real	
ExcessBw	raised cosine filter excess bandwidth	0.5		real	
RecoveryBw	bandwidth of carrier recovery filter	1000	Hz	real	

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

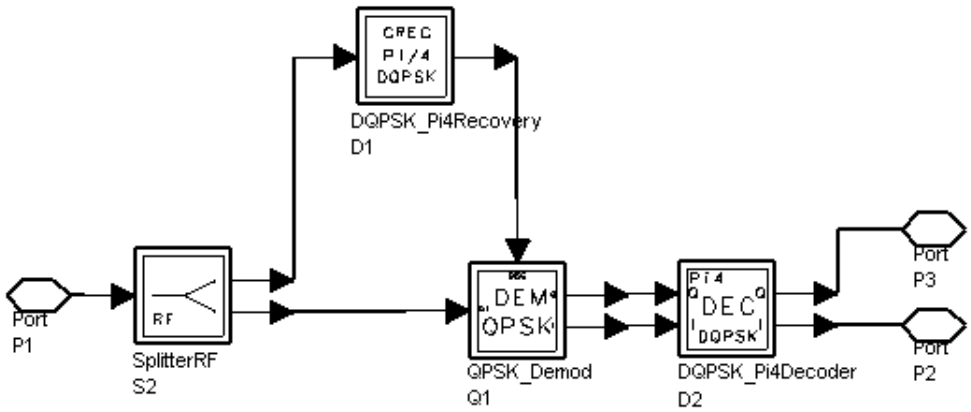
Pin Outputs

Pin	Name	Description	Signal Type
2	I_out	I output	timed
3	Q_out	Q output	timed

Notes/Equations

1. This component is a model of a coherent Pi/4DQPSK demodulator (noncoherent demodulation can be performed with the DQPSK_Pi4Demod component). The demodulator has 1 input pin and 2 output pins. Pin 1 is the input for the Pi/4DQPSK modulated signal; pins 2 and 3 are the output pins for the inphase and quadrature data signals. The schematic for this demodulator is shown below.

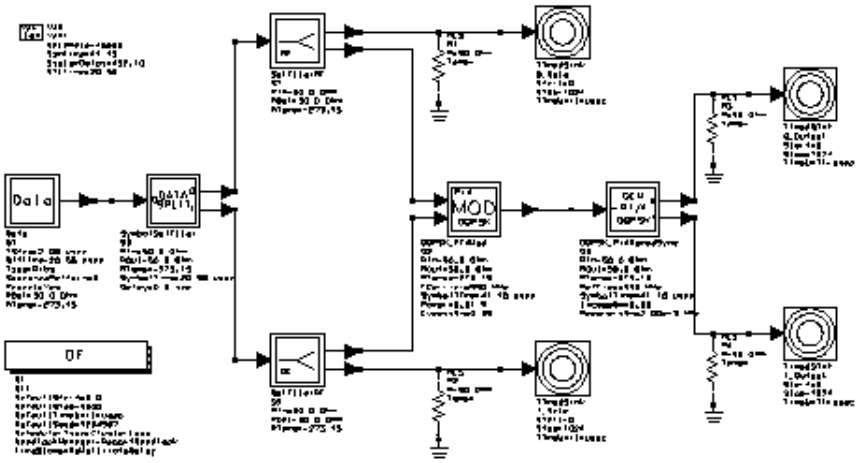
Coherent Pi/4DQPSK Demodulator



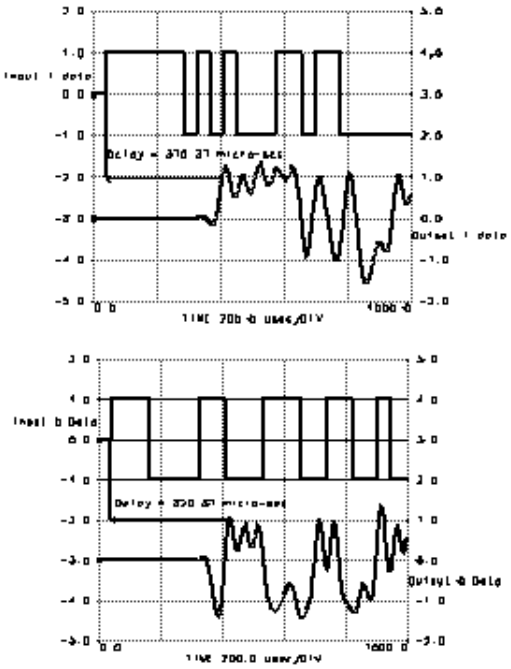
The demodulator uses the DQPSK_Pi4Recovery component to recover a reference carrier signal from the input signal. The modulated signal and the reference carrier signal are then fed to the DQPSK_Demod component, which recovers the inphase and quadrature components and filters them with the appropriate square root raised-cosine filters. These signals are then passed through a differential decoder, which decodes the Pi/4DQPSK differential encoding to produce the required binary data signals.

2. No filter is provided at the input to this demodulator. When noise is present in the channel, an appropriate filter can be placed at the demodulator input.
3. Pi/4DQPSK System: fc=990 MHz, Data rate = 48.6 kbits/sec shows a setup with a modulator and demodulator back-to-back.
   The input and output data streams of the I and Q data channels are shown in Pi/4DQPSK System: fc=990 MHz, Data rate = 48.6 kbits/sec. Note the delay between the input and output data signals. This delay is due to the filters in the modulator and demodulator, each of which introduces a delay of 4SymbolTime seconds, and the differential decoder, which introduces a delay of 1SymbolTime second. Therefore, the delay from input to output is 9SymbolTime seconds, which for a data rate of 48.6 kbits/sec equals 370.37 seconds.
   The eye patterns of the output data signals are shown in Eye Pattern of I and Q Data Signals (carrier recovered using the DQPSK_Pi4Recovery component): fc=990 MHz, Data rate = 48.6 kbits/sec.
4. For certain applications, the I and Q channel data streams may have to be combined into a single data stream. The user must design a clock recovery circuit to generate a clock, or else use the clock component with the appropriate phase to sample the I and Q data streams and combine them using the BinaryCombinerClocked component. As an example, a simple setup to measure the BER of the system is shown in Pi/4DQPSK System: fc=990 MHz, Data rate = 48.6 kbits/sec.
   Additive white Gaussian noise is introduced in the channel. A bandpass filter is placed at the input to the demodulator that lets the signal through without distortion, but removes the out-of-band noise. In Pi/4DQPSK System: fc=990 MHz, Data rate = 48.6 kbits/sec a Clock component is used to generate the sampling clock signal. The phase of the clock is adjusted by the Delay parameter of the Clock component. This phase is adjusted by measuring the delay from the Data component to the output of the DQPSK_Pi4DemodSync component; refer to *Delay Through Pi/4DQPSK System*. The clock is delayed by the total delay plus an additional time of 0.5SymbolTime to sample the data at the midpoint of the symbol. The input and output data streams are shown in Pi/4DQPSK System: fc=990 MHz, Data rate = 48.6 kbits/sec (the input data stream is delayed to align the two data streams).

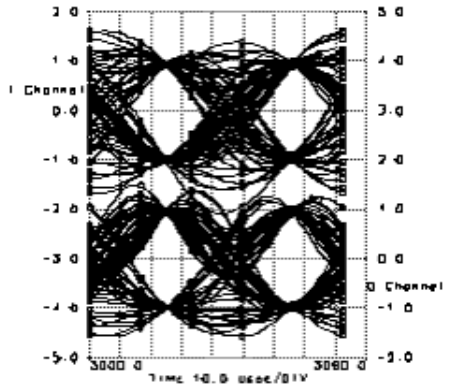
Pi/4DQPSK System: *f* c=990 MHz, Data rate = 48.6 kbits/sec



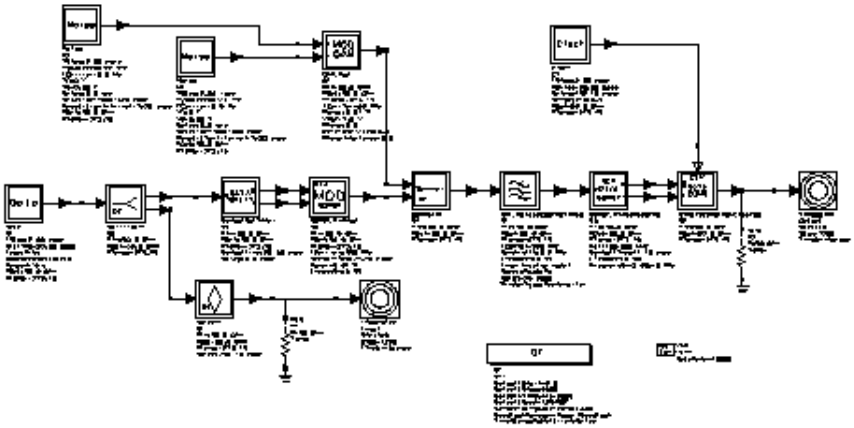
Pi/4DQPSK System: *f* c=990 MHz, Data rate = 48.6 kbits/sec



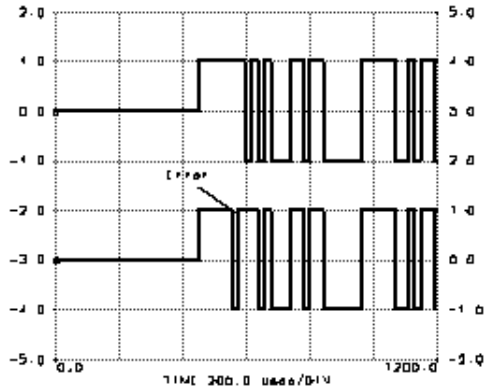
Eye Pattern of I and Q Data Signals (carrier recovered using the DQPSK_Pi4Recovery component): *f* c=990 MHz, Data rate = 48.6 kbits/sec



Pi/4DQPSK System: *f* c=990 MHz, Data rate = 48.6 kbits/sec



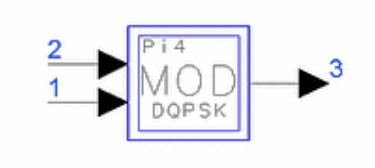
Pi/4DQPSK System: *f* c=990 MHz, Data rate = 48.6 kbits/sec



Delay Through Pi/4DQPSK System

Component	Delay
SymbolSplitter	1 SymbolTime
DQPSK_Pi4Mod	4 SymbolTime
BPF_RaisedCosineTimed	20 µsec
DQPSK_Pi4DemodSync	5 SymbolTime
Total Delay	10 SymbolTime + 20 µsec

DQPSK_Pi4Mod



Description: pi/4-DQPSK modulator with internal oscillator
Library: Timed, Modem
Class: TSDFDQPSK_Pi4Mod
Derived From: baseSubcircuit

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
FCarrier	carrier frequency	1000000	Hz	real	(0, ∞)
SymbolTime	input symbol time interval	0.001	sec	real	[TStep, ∞)†
Power	modulator output power	0.01	W	real	[0, ∞)
ExcessBw	raised cosine filter excess bandwidth	0.5		real	[newpro:0, 1]

† TStep is the simulation time step for the component input signals.

Pin Inputs

Pin	Name	Description	Signal Type
1	I_in	I input	timed
2	Q_in	Q input	timed

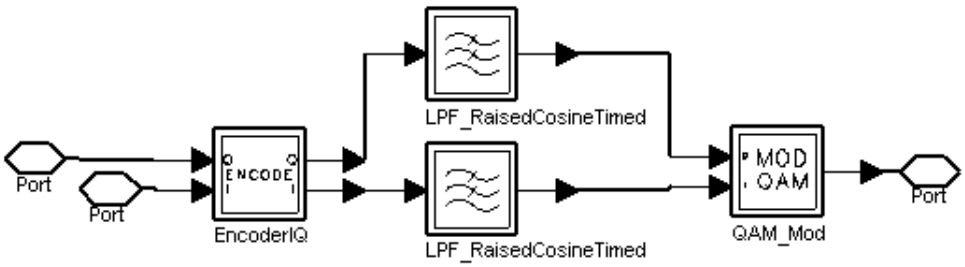
Pin Outputs

Pin	Name	Description	Signal Type
3	output	output signal	timed

Notes/Equations

1. This component is a hierarchical model composed of other components. Pi/4DQPSK Modulator Schematic shows the implementation of the network representing the component.

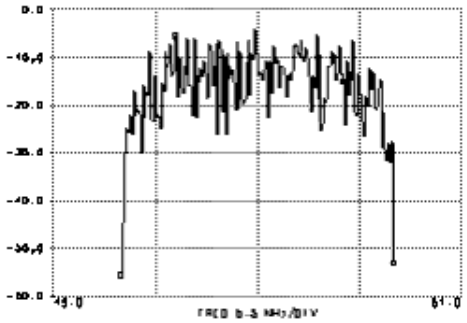
Pi/4DQPSK Modulator Schematic



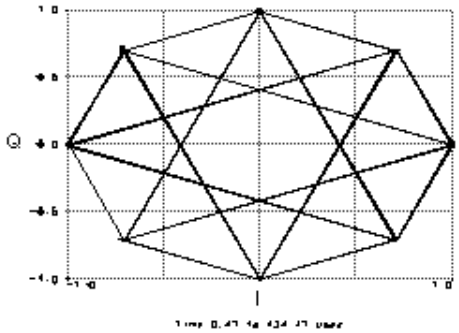
The two inputs to the modulator are the I and Q data streams in NRZ format. The NRZ data symbols are differentially encoded for the Pi/4DQPSK format, filtered with raised-cosine filters, and then modulated to the carrier frequency with a QAM modulator. The following points should be noted about the transmitter.

- The bit time of the input data is specified by the parameter SymbolTime; the roll-off factor of the filters is set by the parameter ExcessBw; and, the carrier frequency and power of the output signal are set by the parameters FCarrier and Power, respectively.
- One-half of the raised-cosine filtering is provided in the transmitter (by using the square root of the cosine filters); the remaining half of the filtering should be performed at the receiver.
- The filters have an (f/sin(f)) shaped equalization to account for the finite width of the input NRZ data.
- The raised-cosine filters have a corner frequency of 1/(2SymbolTime) Hz.
- The impulse response of the filters has a delay of 4SymbolTime seconds. This delay should be increased to 8SymbolTime or greater for better accuracy, especially when making measurements such as EVM.
  The spectrum of the modulated signal is shown in Output Spectrum (Power (dBm) versus Freq) of Pi/4DQPSK Modulator for the following values of the parameters:
  SymbolTime = 1 µsec
  FCarrier = 50 MHz
  Power = 10 dBm
  ExcessBw = 0.35
  The IQ constellation diagram of the modulated signal, after filtering with a bandpass square root raised-cosine filter, is shown in Pi/4DQPSK Constellation Diagram.
  The eye diagram of the I-channel of the modulated signal, after filtering with a bandpass square root raised-cosine filter, is shown in Pi/4DQPSK Eye Diagram.

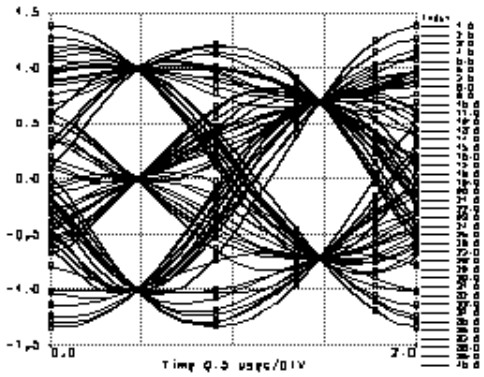
Output Spectrum (Power (dBm) versus Freq) of Pi/4DQPSK Modulator



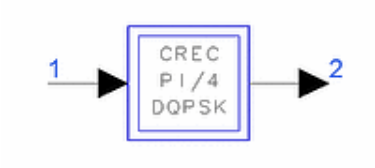
Pi/4DQPSK Constellation Diagram



Pi/4DQPSK Eye Diagram



DQPSK_Pi4Recovery



Description: pi/4-DQPSK carrier recovery
Library: Timed, Modem
Class: TSDFDQPSK_Pi4Recovery
Derived From: baseSubcircuit

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
RefFreq	internal reference frequency	1000000	Hz	real	(0, ∞)
SymbolTime	input symbol time interval	0.001	sec	real	[newpro:TStep, ∞)†
RecoveryBw	bandwidth of carrier recovery filter	1000	Hz	real	(0, ∞)

† TStep is the simulation time step for the component input signal.

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

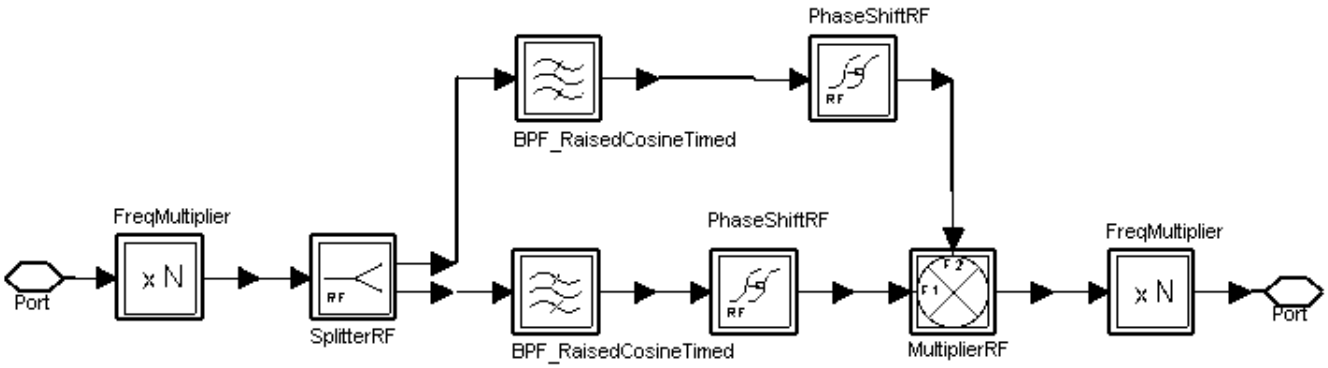
Pint Outputs

Pin	Name	Description	Signal Type
2	output	output signal	timed

Notes/Equations

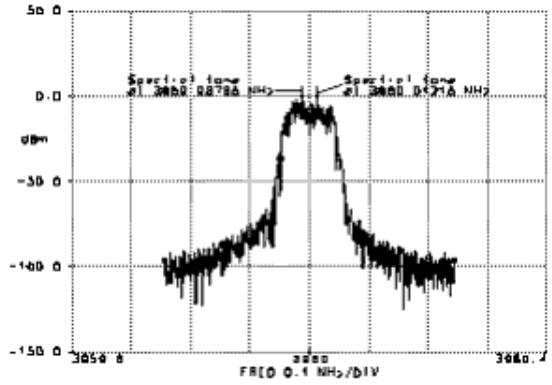
1. This component is a network built up from other components as shown in DQPSK_Pi4Recovery: Carrier Recovery Network. This component recovers a carrier from a modulated Pi/4DQPSK signal, which can then be used for coherent demodulation (refer to the DQPSK_Pi4DemodSync component for more details). In DQPSK_Pi4Recovery: Carrier Recovery Network let
   fc = RefFreq
   STime = 1/Data rate = SymbolTime
   BW = RecoveryBW

DQPSK_Pi4Recovery: Carrier Recovery Network



2. The first step in recovering a carrier is to raise the modulated signal to its fourth power. This is done with the FreqMultiplier component. The spectrum of the signal at the output of the FreqMultiplier component is centered at four times the input carrier frequency and has two distinct tones at
$4 f_c \pm 1(1/(2STIME))$ as shown in Spectrum (Power(dBm) versus Freq) of Fourth Power of Pi/4DQPSK Signal: fc=990 MHz, Data rate = 48.6 kbits/sec.
The two tones at $4 f_c \pm 1(1/(2STIME))$ and $4 f_c - 1(1/(2STIME))$ are filtered and recombined to produce the required carrier at f_c. The PhaseShiftRF components compensate for the phase shift introduced due to the delay through the filters.

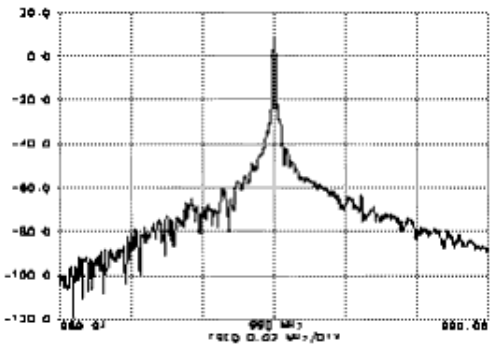
Spectrum (Power(dBm) versus Freq) of Fourth Power of Pi/4DQPSK Signal: *f* c=990 MHz, Data rate = 48.6 kbits/sec



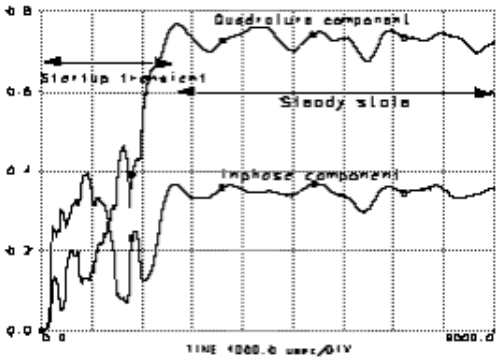
3. The recovered carrier signal will have a phase ambiguity of 45 degrees. This is because of the divide-by-8 component (the FreqMultiplier component with X=0.125).
4. RecoveryBw specifies the bandwidth of the filters (these bandpass filters can be replaced by PLLs). The spectral purity of the recovered carrier increases as the bandwidth decreases; however, the initial transient time increases. Since the two spectral tones are spaced at 1/SymbolTime Hz apart, Bw should be much smaller than this value.
Spectrum (Power(dBm) versus Freq) of Recovered Carrier Signal, Modulation = Pi/4DQPSK: fc=990 MHz, Data rate = 48.6 kbits/sec, BW=20 kHz shows the spectrum of the recovered carrier (the spectrum shown is of the carrier at steady state). Inphase and Quadrature Phase Components of Recovered Carrier Signal, Modulation = Pi/4DQPSK, fc=990 MHz, Data rate = 48.6 kbits/sec, BW=20 kHz shows the inphase and quadrature components of the recovered carrier signal and the start-up transient.

5. For certain applications, the filter bandwidths required to extract a clean reference signal may be prohibitively small. Further, the simulation time will become very large (due to the large number of taps required to model the filter), and the initial start-up transient will last for a long time. In such cases it will be better to use the N_Tones component to directly generate the carrier.

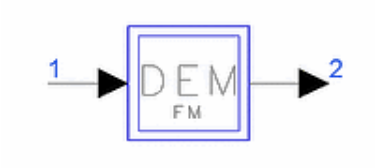
Spectrum (Power(dBm) versus Freq)of Recovered Carrier Signal, Modulation = Pi/4DQPSK: *f* c=990 MHz, Data rate = 48.6 kbits/sec, BW=20 kHz



Inphase and Quadrature Phase Components of Recovered Carrier Signal, Modulation = Pi/4DQPSK, *f* c=990 MHz, Data rate = 48.6 kbits/sec, BW=20 kHz



FM_Demod



Description: Frequency demodulator with internal oscillator
Library: Timed, Modem
Class: TSDF_FM_Demod
Derived From: baseOmniSysStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
RefFreq	internal reference frequency	1000000	Hz	real	{-1} or (0, ∞)†
Sensitivity	voltage output sensitivity, Vout/Hz	1		real	(-∞, ∞)
Phase	initial phase of carrier, in degrees	0	deg	real	(-∞, ∞)

† Refer to Note 2 for the special meaning of a -1 value.

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
2	output	output signal	timed

Notes/Equations

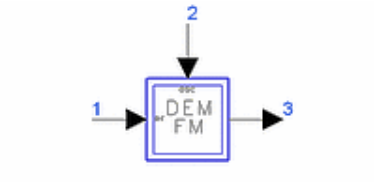
1. The inphase and quadrature components I(t) and Q(t) of the modulated signal are extracted according to the procedure described in *Timed Modem Components* (timed). The output at pin 2 is then calculated as

$$V_2(t) = \frac{S}{2 \times \pi} \times \frac{d}{dt}\left(\operatorname{atan}\left(\frac{Q(t)}{I(t)} \right) \right)$$

2. When RefFreq = −1, then internal oscillator frequency synchronization to the input signal is performed as described in *Timed Modem Components* (timed). This synchronization is only allowed when the input signal is an RF (complex envelope) timed signal.
3. When the input signal is a baseband signal, input signal transformation to an RF (complex envelope) timed signal is performed as described in *Demodulation*

Algorithms (timed), case 2, before the demodulation process is performed.

FM_DemodExtOsc



Description: Frequency demodulator with external reference oscillator
Library: Timed, Modem
Class: TSDF_FM_DemodExtOsc
Derived From: baseOmniSysStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
Sensitivity	voltage output sensitivity, Vout/Hz	1		real	(-∞, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed
2	osc	RF oscillator signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
3	output	output signal	timed

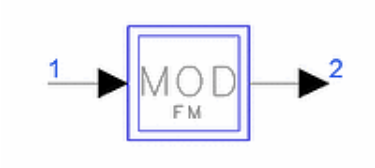
Notes/Equations

1. The inphase and quadrature components I(t) and Q(t) of the modulated signal are extracted according to the procedure described in *Timed Modem Components* (timed). The output at pin 3 is then calculated as

$$V_3(t) = \frac{S}{2 \times \pi} \times \frac{d}{dt}\left(\operatorname{atan}\left(\frac{Q(t)}{I(t)}\right)\right)$$

2. When the input signal is a baseband signal, input signal transformation to an RF (complex envelope) timed signal is performed as described in *Demodulation Algorithms* (timed), case 2, before the demodulation process is performed.

FM_Mod



Description: Frequency modulator with internal oscillator
Library: Timed, Modem
Class: TSDF_FM_Mod
Derived From: baseOmniSysStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
FCarrier	carrier frequency	1000000	Hz	real	(0, ∞)
Power	unmodulated carrier power	0.01	W	real	[0, ∞)
Sensitivity	frequency deviation sensitivity, Hz/volts	1		real	(-∞, ∞)
Phase	phase reference in degrees	0	deg	real	(-∞, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
2	output	output signal	timed

Notes/Equations

1. This component is a frequency modulator whose modulation frequency, output power and modulation index can be set by the user.
2. The output of an FM modulator can be expressed as:

$$V_2(t) = A\cos\left(2\pi f_c t + 2\pi s\int_0^t V_1(\alpha)d\alpha\right)$$

where s equals Sensitivity. In the program, the signal $V_2(t)$ is represented in terms of its inphase and quadrature components.

$$V_2(t) = Re\left\{(v_2)e^{j\omega_c t}\right\}$$

$$v_2(t) = (v_{I2}(t) + jv_{Q2}(t))$$

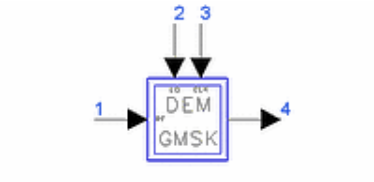
$$v_2(t) = Ae^{j\omega_c t + P + 2\pi s\int_0^t (v_1(\alpha)d\alpha)} e^{(-j)\omega_c t}$$

$$A = \sqrt{(2)(ROut)10^{\frac{PdBm-30}{10}}}, \quad \omega_c = 2\pi f_c$$

where
PdBm = power in dBm units
P = πPhase/180
f_c = FCarrier

3. Ensure that the output of FM_Mod is sampled at a sufficiently high rate- this is because of the bandwidth expansion that typically takes place when a signal is frequency modulated. The bandwidth of the modulated output can be estimated using Carson's rule
$$B \approx 2(f_{max} + Sv_{max})$$
where f_{max} is the highest frequency component of the input signal and v_{max} is the maximum amplitude of the input signal. The sampling period T_S of the output signal should then satisfy the condition $T_S << 1/B$. For most applications $T_S = 1/(5B)$ should be a sufficiently small sampling period.

GMSK_Demod



Description: GMSK demodulator with external reference oscillator
Library: Timed, Modem
Class: TSDFGMSK_Demod
Derived From: baseSubcircuit

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed
2	LO	RF local oscillator signal	timed
3	clock	clock signal	timed

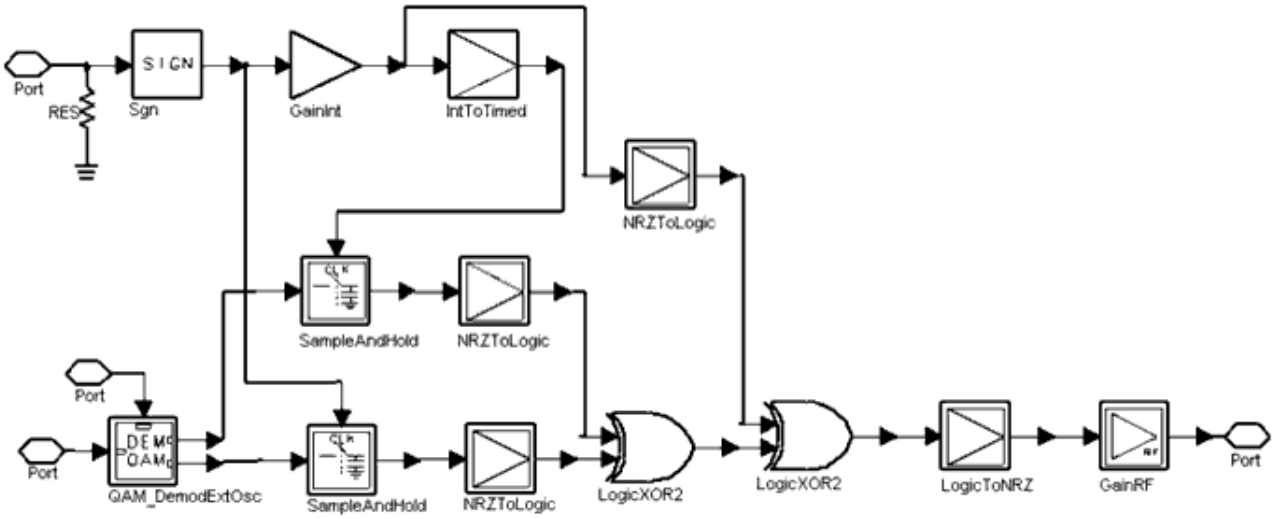
Pin Outputs

Pin	Name	Description	Signal Type
4	output	output signal	timed

Notes/Equations

1. GMSK_Demod is a model of a suboptimal (an optimal receiver would require the Viterbi algorithm) coherent GMSK demodulator (refer to the GMSK_Recovery component). Noncoherent demodulation can be performed with the FM_Demod component.
2. Pin 1 is the input for the GMSK modulated RF signal; pin 2 is the input for a reference RF carrier signal used to demodulate the GMSK signal and obtain its inphase and quadrature components; pin 3 is the input for a clock baseband signal used to sample the demodulated inphase and quadrature signals.
   The reference RF carrier signal should be synchronized in frequency and phase to the modulated signal for proper demodulation. The clock signal should have a duty cycle of 50 percent; its frequency should equal half the transmitted data rate; and, its phase should be adjusted so that the positive edge of the clock occurs at the instant when the eye of the demodulated inphase signal is maximum (therefore, the negative edge will occur when the eye of the demodulated quadrature signal is maximum). Coherent GMSK Demodulator Schematic shows the demodulator schematic.

Coherent GMSK Demodulator Schematic



3. To understand the operation of this demodulator, note that GMSK is a digital phase modulation technique where the phase of the carrier φ(*t*), changes by ±π/2 radians (approximately) in a symbol period, depending on whether the input is ±1.
Let *I(t)* and *Q(t)* be the inphase and quadrature components of the GMSK signal. Let T_S be the duration of an data symbol, and let 2 *nT* S, n=0,1,2, ... be the instances when *I(t)* is sampled and (2 *n+1)T* S, n=0,1,2, ... be the instances when *Q(t)* is sampled. The samples of *I(t)* and *Q(t)* then provide the following information about the phase of the modulated signal:

$$\phi(2nT_s) = \begin{cases} 0 & I(2nT_s) > 0 \\ \pi & I(2nT_s) < 0 \end{cases}$$

$$\phi((2n+1)T_s) = \begin{cases} \pi/2 & Q((2n+1)T_s) > 0 \\ -\pi/2 & Q((2n+1)T_s) < 0 \end{cases}$$

By following the phase trajectory, the information sequence can be determined. In the demodulator circuit shown in Coherent GMSK Demodulator Schematic, the demodulated inphase and quadrature signals are sampled alternately and samples $I(2nT_S)$ and $Q((2n+1)T_S)$ are stored in the sample and hold components. The logic gates then decode the information sequence X_n, n = 0,1, ... from the samples based on

$$X_n = \begin{cases} 0 & \phi((n+1)T_s) > \phi(nT_s) \\ 1 & \phi((n+1)T_s) < \phi(nT_s) \end{cases}$$

4. The following points should be noted about this demodulator.
   - A filter is not provided; an appropriate filter can be placed at the input.
   - The carrier and clock signal can be recovered from the modulated signal with the GMSK_Recovery component. Or, ideal signals can be generated using N_Tones and Clock components, in which case the carrier and clock signals

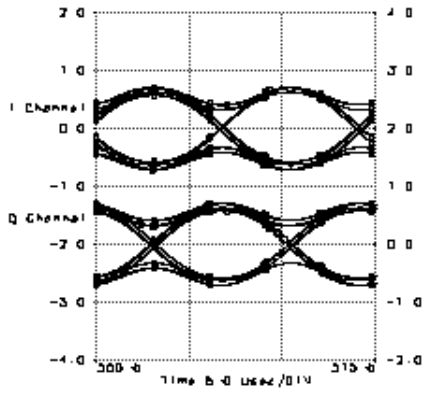
should be synchronized with the received modulated signal.

- Eye pattern of Demodulated Inphase and Quadrature Components (Carrier Recovered using GMSK_Recovery Component) Modulation=0.3GMSK, fc=990MHz, STime=3.7µsec shows the eye patterns of the demodulated (i.e., at the outputs of the QAM_Demod component within the GMSK_Demod component) inphase and quadrature components of a 0.3GMSK signal. Demodulated Inphase and Quadrature Signals with Clock Obtained from the GMSK_Recovery Component Modulation=0.3GMSK, fc=990MHz, STime=3.7µsec shows the demodulated inphase and quadrature signals along with the clock signal. Note how the positive edge of the clock can be used to sample the inphase signal and the negative edge of the clock can be used to sample the quadrature signal.
  The input and the demodulated data streams are shown in Input DATA and Output Data Signals Modulation=0.3GMSK, fc=990MHz, STime=3.7µsec. Demodulated data is delayed 11.1 µsec with respect to the input data stream (the GMSK modulator introduces a delay of 0.6/ *F* 3dB=7.4µsec and the demodulator introduces a delay of 1 symbol time or 3.7 µsec, for a total of 11.1 µsec).
- The BER performance of this demodulator for an additive Gaussian channel is given in *Measured BER Performance for 0.3GMSK (with Carrier and Clock Recovery) in an AWGN Channel* (these BER measurements were made using Monte Carlo simulations). Additive white Gaussian noise was introduced in the channel and an ideal bandpass filter was placed at the receiver input. The filter bandwidth was first optimized by varying the bandwidth at a fixed SNR till the BER was minimized.

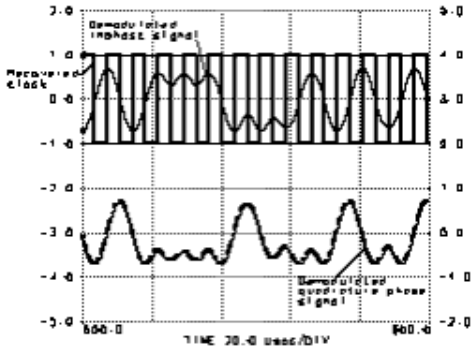
Measured BER Performance for 0.3GMSK (with Carrier and Clock Recovery) in an AWGN Channel

Eb/No (dB)	BER
4.1	1.0×10^{-3}
4.5	1.22×10^{-3}
4.9	7.8×10^{-4}
5.3	4.6×10^{-4}

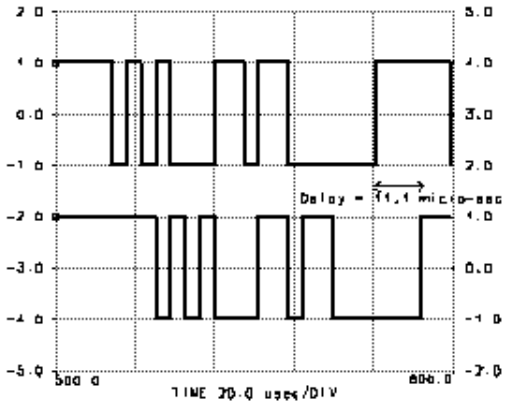
Eye pattern of Demodulated Inphase and Quadrature Components (Carrier Recovered using GMSK_Recovery Component) Modulation=0.3GMSK, f_c = 990MHz, *STime* = 3.7µsec



Demodulated Inphase and Quadrature Signals with Clock Obtained from the GMSK_Recovery Component Modulation=0.3GMSK, f_c = 990MHz, *STime* = 3.7µsec



Input DATA and Output Data Signals Modulation=0.3GMSK, f_c = 990MHz, *STime* = 3.7µsec



S

GMSK_Mod

Description: GMSK modulator with internal oscillator
Library: Timed, Modem
Class: TSDFGMSK_Mod
Derived From: baseSubcircuit

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0,)
ROut	output resistance	DefaultROut	Ohm	real	(0,)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15,)
FCarrier	carrier frequency	1000000	Hz	real	(0,)
SymbolTime	input symbol time interval	0.001	sec	real	[TStep,)†
F3dB	3dB frequency of input Gaussian lowpass filter	500000	Hz	real	(0,)
Power	modulator output power	0.01	W	real	[0,)

† TStep is the simulation time step for the component input signal.

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

Pin Ouputs

Pin	Name	Description	Signal Type
2	output	output signal	timed

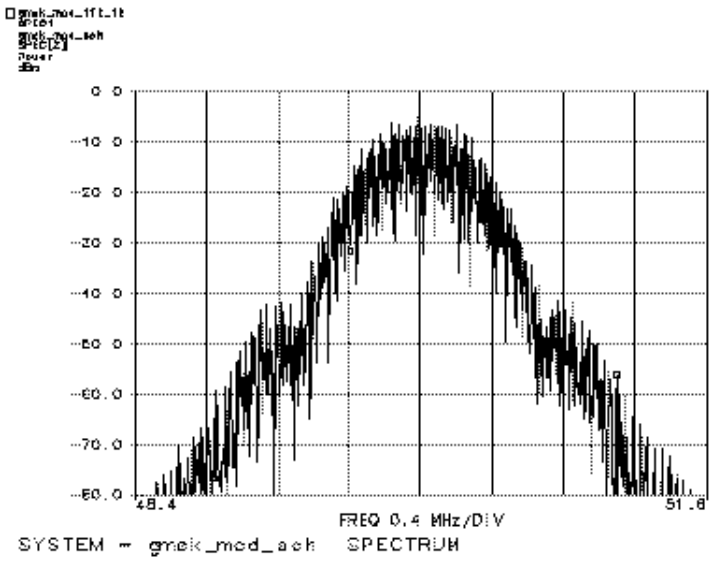
Notes/Equations

1. This component is a hierarchical model composed of other components. GMSK Modulator Schematic shows the implementation of the network representing the component.
   In this implementation of the GMSK modulator, the input data is filtered by a Gaussian filter and then input to an FM modulator with a modulation sensitivity of 1/(4SymbolTime) Hz/volt. The input bits are first hard-limited to ensure that they are in the NRZ format. The bit time of the input data is specified by the parameter SymbolTime; the carrier frequency and power of the output signal are set by the parameters FCarrier and Power, respectively. The 3-dB bandwidth of the Gaussian filter is set by the parameter F3dB. A common choice for the 3-dB bandwidth is given

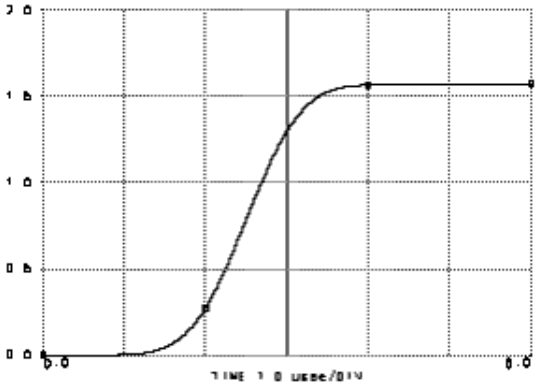
by the relation F3dB=0.3/SymbolTime, and this scheme is known as 0.3GMSK. The spectrum of the modulated signal is shown in Output Spectrum (Power(dBm) versus Freq) of GMSK Modulator for 0.3GMSK and the following values of the parameters:
SymbolTime = 1 µsec, F3dB = 0.3 MHz,
FCarrier = 50 MHz, Power = 10 dBm.
The phase response of the modulated signal of a 0.3GMSK modulator for an input pulse of 1 µsec duration and amplitude of 1V is shown in Phase Response of GMSK Modulator. The vertical axis displays the phase in radians. Note that it takes approximately 4 µsec for the phase to reach its final value of /2 radians.

GMSK Modulator Schematic

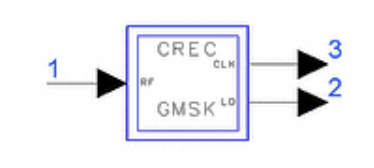
Output Spectrum (Power(dBm) versus Freq) of GMSK Modulator



Phase Response of GMSK Modulator



GMSK_Recovery



Description: GMSK carrier recovery
Library: Timed, Modem
Class: TSDFGMSK_Recovery
Derived From: baseSubcircuit

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
RefFreq	internal reference frequency	1000000	Hz	real	(0, ∞)
SymbolTime	input symbol time interval	0.001	sec	real	[TStep, ∞)†
RecoveryBw	bandwidth of carrier recovery filter	1000	Hz	real	(0, ∞)

† TStep is the simulation time step for the component input signal.

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
2	LO	RF local oscillator signal	timed
3	clock	clock signal	timed

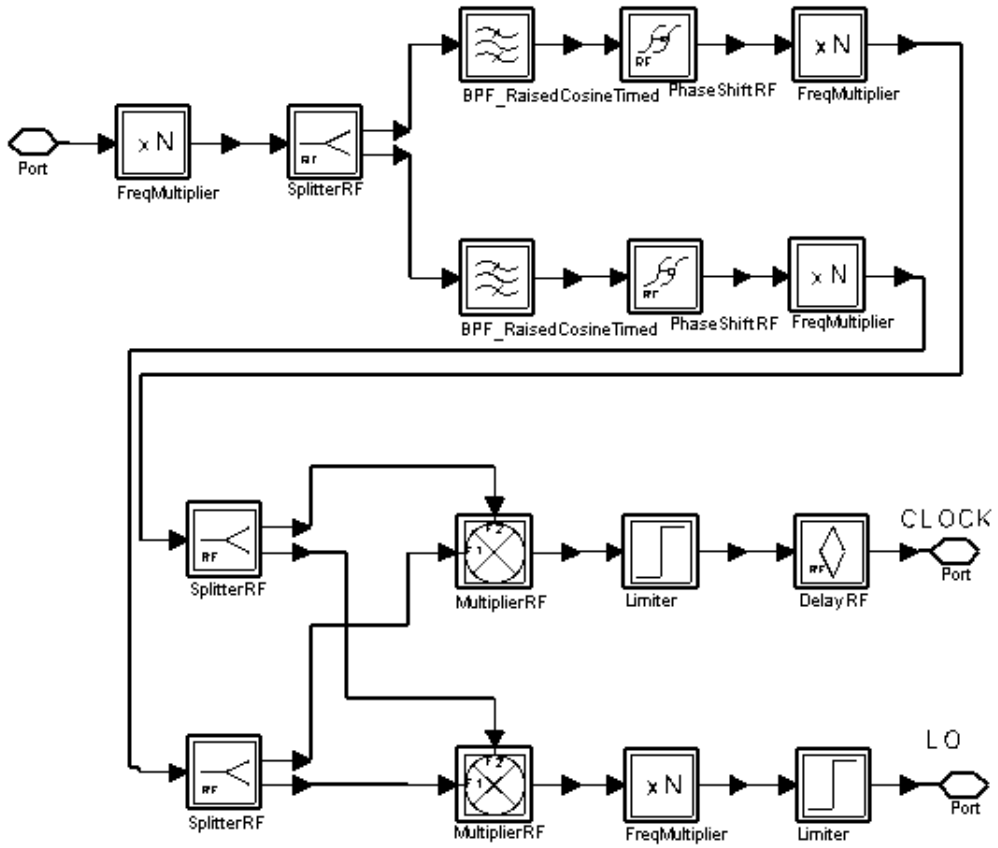
Notes/Equations

1. GMSK_Recovery is a network built from other components as shown in GMSK_Recovery Clock/Carrier Recovery Network. It recovers a carrier and clock signal from a modulated GMSK signal, which are required for coherent demodulation of the GMSK signal (refer to the GMSK_Demod component for more details).
2. The first step in recovering a clock and carrier is to square the input signal-this is done with the FreqMultiplier component. The spectrum of the signal at the output of the FreqMultiplier component is centered at twice the input carrier frequency and has tones offset from 2 f_c at odd multiples of 1/(2SymbolTime) as shown in Spectrum (Power(dBm) versus Freq) of Squared 0.3GMSK Signal, fc =990MHz, STime=3.7µsec. The two tones at 2 f_c + (1/(2SymbolTime)) and 2 f_c - (1/(2SymbolTime)) are

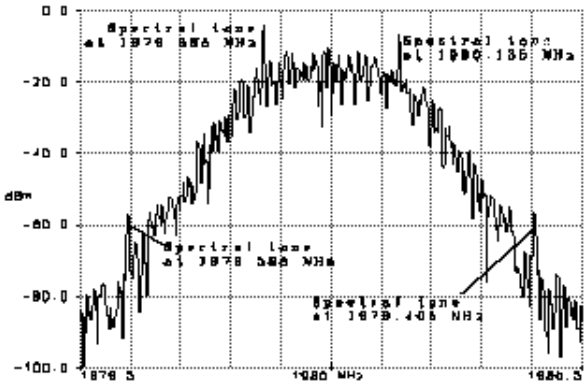
filtered and recombined to produce the required carrier at f_c and a clock of period SymbolTime. The PhaseShiftRF component compensates for the phase shift introduced due to the delay through the filters.

3. Pins 1 and 3 (the RF input pin and the reference carrier output pin) have impedances while pin 2 (the clock output) has a 0 ohm impedance.
4. The recovered carrier signal can have a phase ambiguity of 90 degrees. Phase Trajectory of Recovered Carrier Signal, Modulation = 0.3GMSK, fc = 990MHz, SymbolTime = 3.7µsec, Bw = 10kHz shows the phase trajectory of the recovered carrier when the carrier recovery circuit is attached directly to the output of the GMSK modulator (the phase shown in Phase Trajectory of Recovered Carrier Signal, Modulation = 0.3GMSK, fc = 990MHz, SymbolTime = 3.7µsec, Bw = 10kHz is of the recovered carrier before it is passed through the hard limiter). The phase of the carrier is finally 180 degrees rather than 0 degrees.
5. The parameter RecoveryBw specifies the bandwidth of the filters (these bandpass filters can be replaced by PLLs). The spectral purity of the recovered carrier increases as the bandwidth decreases. However, the initial transient time increases. Since the two spectral tones are spaced at 1/SymbolTime Hz apart, Bw should be much smaller than this value. Spectrum (Power(dBm) versus Freq) of Recovered Carrier Signal, Modulation = 0.3GMSK, fc = 990MHz, SymbolTime = 3.7µsec, Bw = 10kHz shows the spectrum of the recovered carrier (the spectrum shown is of the carrier at steady state). Better performance may be obtained by choosing a window other than a rectangular window with the WindowType parameter of the bandpass filters. Inphase and Quadrature Phase Components of Recovered Carrier Signal Modulation = 0.3GMSK, fc = 990MHz, SymbolTime = 3.7µsec, Bw = 10kHz shows the inphase and quadrature components of the recovered carrier signal and the start-up transient.
6. The recovered clock signal has a period of 2SymbolTime sec and a duty cycle of 50 percent. The phase of the clock is such that the positive edge of the clock can be used to sample the I-channel of the demodulator output and the negative edge can be used to sample the Q-channel of the demodulator output (refer to the GMSK_Demod component for more details).
7. For certain applications, the filter bandwidths required to extract a clean reference signal may be prohibitively small. Further, the simulation time will become very large (due to the large number of taps required to model the filter), and the initial start-up transient will last for a long time. In such cases, use the N_Tones component to directly generate the carrier.

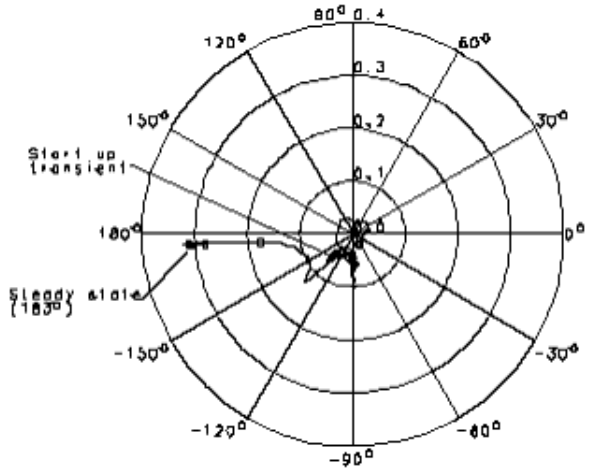
GMSK_Recovery Clock/Carrier Recovery Network



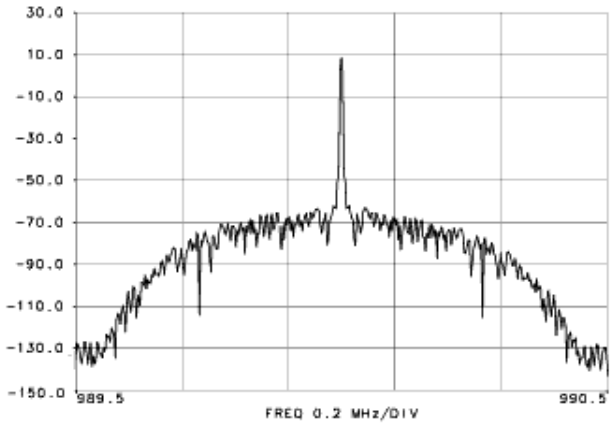
Spectrum (Power(dBm) versus Freq) of Squared 0.3GMSK Signal, f_c = 990MHz, *STime* = 3.7µsec



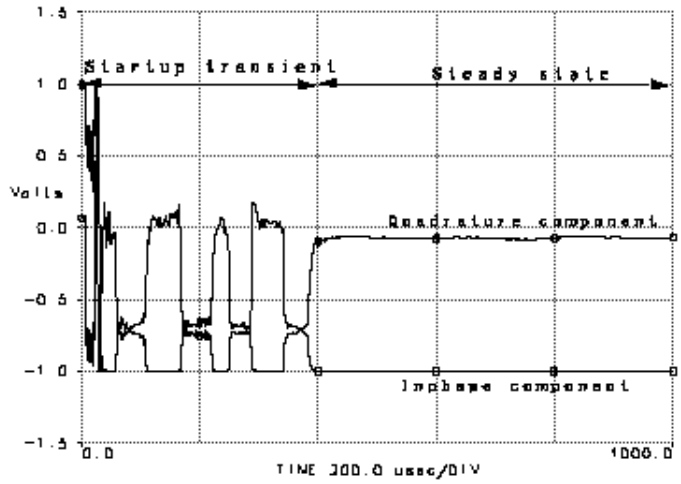
Phase Trajectory of Recovered Carrier Signal, Modulation = 0.3GMSK, f_c = 990MHz, *SymbolTime* = 3.7µsec, *Bw* = 10kHz



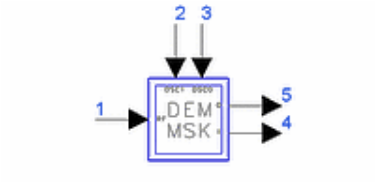
Spectrum (Power(dBm) versus Freq) of Recovered Carrier Signal, Modulation = 0.3GMSK, f_c = 990MHz, *SymbolTime* = 3.7µsec, *Bw* = 10kHz



Inphase and Quadrature Phase Components of Recovered Carrier Signal Modulation = 0.3GMSK, f_c = 990MHz, *SymbolTime* = 3.7µsec, *Bw* = 10kHz



MSK_Demod



Description: MSK demodulator with external reference oscillators
Library: Timed, Modem
Class: TSDFMSK_Demod
Derived From: baseSubcircuit

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed
2	osc_I	I oscillator signal	timed
3	osc_Q	Q oscillator signal	timed

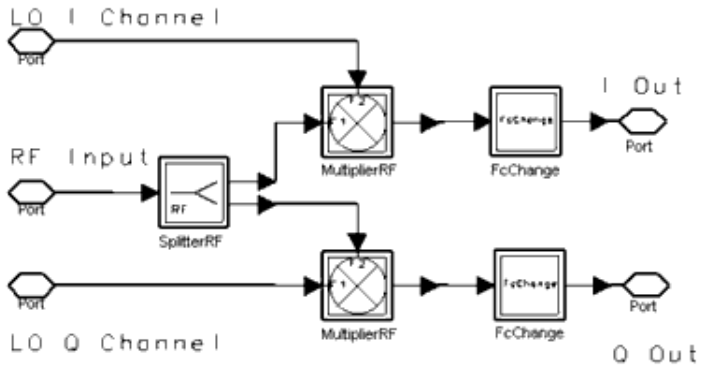
Pin Outputs

Pin	Name	Description	Signal Type
4	I_out	I output	timed
5	Q_out	Q output	timed

Notes/Equations

1. MSK_Demod is composed of other components. MSK_Demod Demodulator Network shows the implementation of the network representing the demodulator.
   The demodulator consists of two down-converters followed by the FcChange component, which forces the output signals to be baseband signals. The inputs to the demodulator are the RF signal and the reference signals OSCI and OSCQ for the I- and Q-channels, respectively. (RF filtering is not included at the input of this MSK demodulator; if required, the appropriate filter must be placed by the user.)

MSK_Demod Demodulator Network

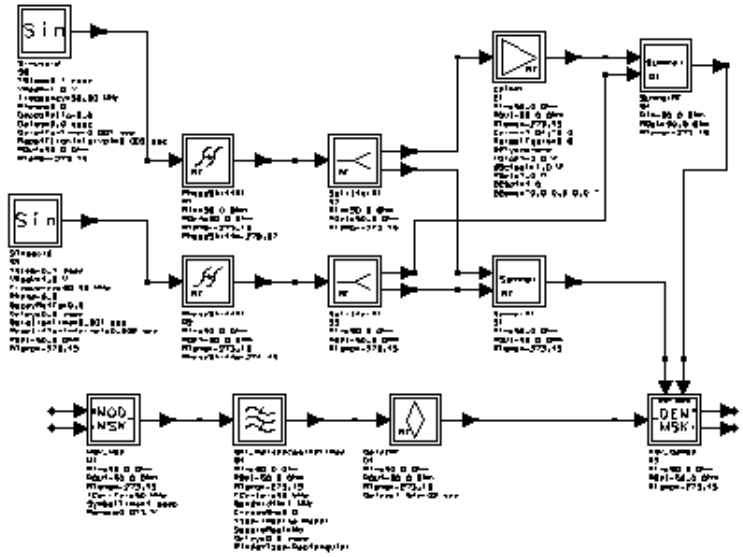


2. The reference signals OSCI and OSCQ that are required for demodulating the I- and Q-channel signals are the product of the RF carrier and the pulse shaping sinusoids:
   OSCI = $\cos(2\pi f_c t + \theta) \cos(\pi t/\text{STIME} + \theta')$

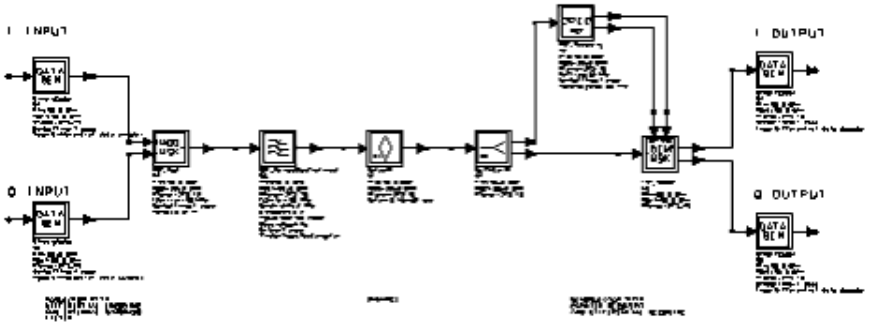
   OSCO = $\sin(2\pi f_c t + \theta) \sin(\pi t/\text{STIME} + \theta')$

   Here f_c is the carrier frequency, SymbolTime is the input bit time at the modulator, and θ and θ' are the phase offsets due to channel delays. The phase of the reference signals must be matched to the phase of the received RF signal for proper demodulation of the MSK signal. If the channel delay is known, reference signals OSCI and OSCQ can be generated locally at the receiver, as shown in MSK Demodulation with Local Reference Signals; if the channel delay is unknown, the MSK_Recovery component can be used to recover the reference signals from the RF signal.
   The recovered reference signals from the MSK_Recovery component have a phase ambiguity of 180 degrees. Therefore, differential encoding and decoding is required with this carrier recovery scheme.
   Differential encoding can be performed by inserting BinaryCoder components in the I- and Q-channels at the input of the modulator. Differential decoding can be performed at the receiver by inserting the BinaryCoder component at the outputs of the I- and Q-channels of the decoder.
   Note that with differential coding and decoding, the first pair of decoded bits can be in error because the starting state of the differential coder is not known by the differential decoder. MSK Demodulation with Carrier Recovery illustrates how such a system can be set up.
   For more details on the MSK demodulator, see S. Pasupathy, "Minimum Shift Keying: A Spectrally Efficient Modulation," *IEEE Communications* , 1979, pp. 14-22.

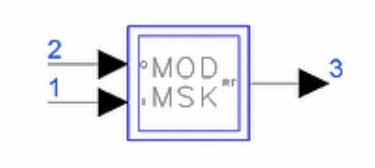
MSK Demodulation with Local Reference Signals



MSK Demodulation with Carrier Recovery



MSK_Mod



Description: MSK modulator with internal oscillator
Library: Timed, Modem
Class: TSDFMSK_Mod
Derived From: baseSubcircuit

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
FCarrier	carrier frequency	1000000	Hz	real	(0, ∞)
SymbolTime	input symbol time interval	0.001	sec	real	[TStep, ∞) †
Power	modulator output power	0.01	W	real	[0, ∞)

† TStep is the simulation time step for the component input signals.

Pin Inputs

Pin	Name	Description	Signal Type
1	I_in	I input	timed
2	Q_in	Q input	timed

Pin Outputs

Pin	Name	Description	Signal Type
3	output	output signal	timed

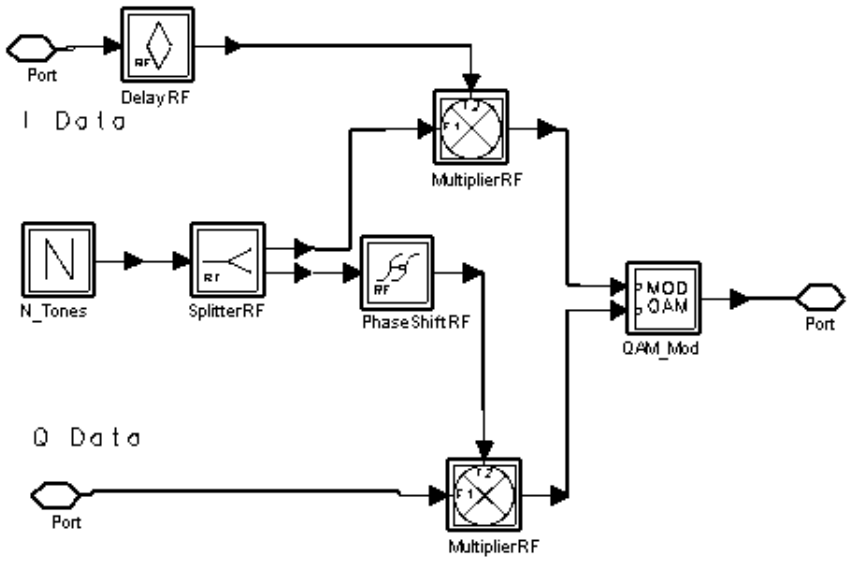
Notes/Equations

1. MSK_Mod is a hierarchical model composed of other components. MSK Modulator Schematic shows the implementation of the network representing the component. This implementation of the MSK modulator is often referred to in literature as a parallel implementation. It is, in fact, a linear modulation scheme applied on the I and Q data streams. The two inputs to the modulator are the I and Q data streams in NRZ format. The data in the Q-channel is first delayed by half a bit period. The I and Q data are then sinusoidally shaped and modulated by a QAM modulator. The bit period of the input data is specified by the SymbolTime parameter; the carrier frequency and power of the output signal are set by FCarrier and Power, respectively. The spectrum of the modulated signal is shown in Output Spectrum of MSK Modulator for the following values:

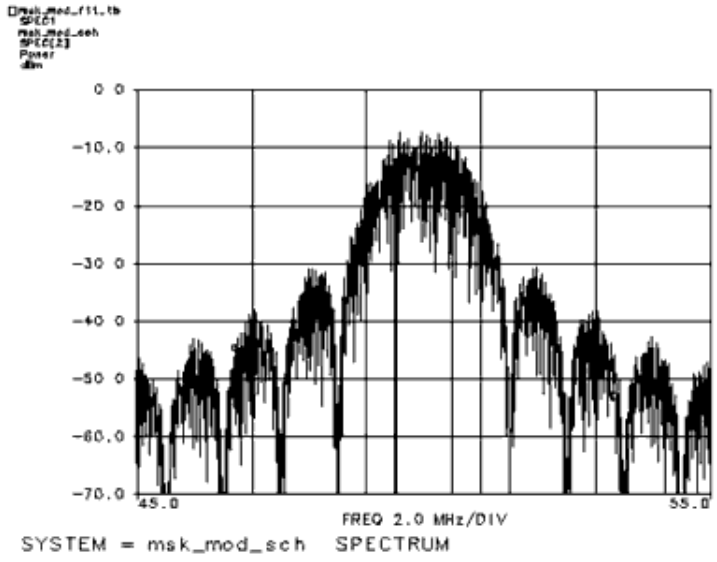
SymbolTime = 1 µsec
FCarrier = 50 MHz
Power = 10 dBm
Because filters are not used in this network, the output signal has a wide spectrum. It is therefore recommended that in any system application the output be filtered by a bandpass filter of the user's choice.
The phase trellis of the modulated signal is shown in Phase (radian) Response of MSK Modulator (the vertical axis displays the phase in radians, which is measured modulo 2π in the range [-π, π]). Note that the phase varies linearly with time and changes by π/2 radians in a time interval of SymbolTime/2 seconds.

2. It is also possible to implement a serial MSK modulator. One implementation is to use an FM modulator with a sensitivity of 1/(4T) Hz/volt for an input data stream of 1/T bits/second in the NRZ data format. This network is shown in Serial MSK Modulator Schematic.

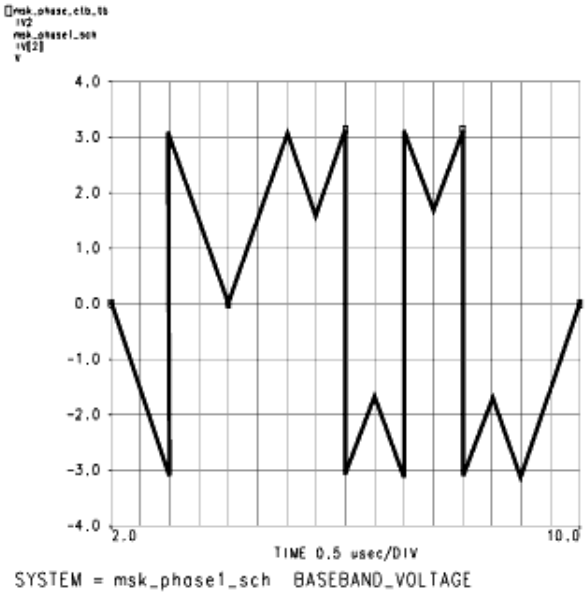
MSK Modulator Schematic



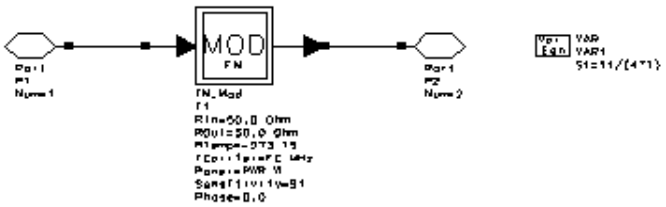
Output Spectrum of MSK Modulator



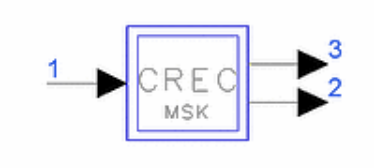
Phase (radian) Response of MSK Modulator



Serial MSK Modulator Schematic



MSK_Recovery



Description: MSK carrier recovery
Library: Timed, Modem
Class: TSDFMSK_Recovery
Derived From: baseSubcircuit

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
RefFreq	internal reference frequency	1000000	Hz	real	(0, ∞)
SymbolTime	input symbol time interval	0.001	sec	real	[newpro:TStep, ∞)†
RecoveryBw	bandwidth of carrier recovery filter	1000	Hz	real	(0, ∞)

† TStep is the simulation time step for the component input signal.

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
2	osc_I	I oscillator signal	timed
3	osc_Q	Q oscillator signal	timed

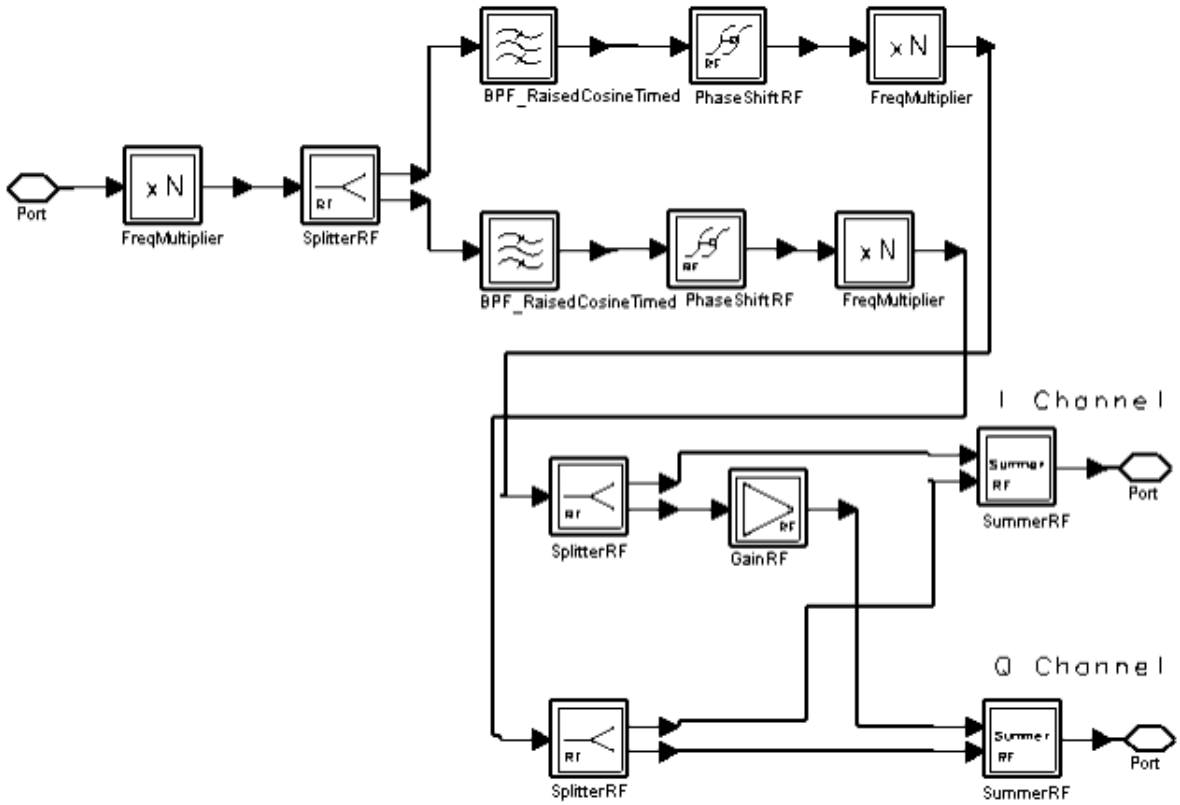
Notes/Equations

1. MSK_Recovery is composed of other components. MSK Carrier Recovery Network shows the implementation of the network representing the MSK carrier recovery circuit. To demodulate the MSK signal, it is necessary to synchronize the inphase and quadrature RF carriers, and the pulse shape cosine and sine waveforms.
   The MSK signal is initially squared by the FreqMultiplier (X=2) component, resulting in two discrete spectral tones at 2(RefFreq + 1/(2 SymbolTime)) and 2(RefFreq − 1/(2 SymbolTime)). These components are filtered by bandpass filters and their frequency is then halved by the FreqMultiplier (X=0.5) components to produce two carriers at (RefFreq + 1/(2 SymbolTime)) and (RefFreq − 1/(2 SymbolTime)).
   The resulting signals are passed through a hard limiter to restore their amplitude.

The sum and difference of the signals then produces the required reference carriers (for details, see Reference 1).

2. Note that the filters have a delay of 4/RecoveryBw. The phase shift components PhaseShiftRF compensate for the phase shift introduced by this delay. The spectral purity of the recovered carrier increases as the bandwidth of the filter is decreased-however, the delay through the circuit increases proportionally. Better performance may be obtained by choosing a window other than a rectangular window with the WindowType parameter of the bandpass filters.

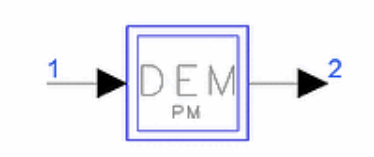
MSK Carrier Recovery Network



References

1. S. Pasupathy, "Minimum Shift Keying: A Spectrally Efficient Modulation," *IEEE Communications*, 1979, pp. 14-22.

PM_Demod



Description: phase demodulator with internal oscillator
Library: Timed, Modem
Class: TSDF_PM_Demod
Derived From: baseOmniSysStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
RefFreq	carrier frequency	1000000	Hz	real	{-1} or (0, ∞)†
Sensitivity	voltage output sensitivity, Vout/degree	1		real	(-∞, ∞)
Phase	phase reference in degrees	0	deg	real	(-∞, ∞)

† Refer to Note 2 for the special meaning of a -1 value.

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
2	output	output signal	timed

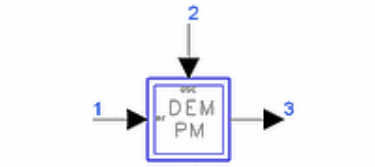
Notes/Equations

1. The inphase and quadrature components I(t) and Q(t) of the modulated signal are extracted according to the procedure described in *Timed Modem Components* (timed). The output at pin 2 is then calculated as:

$$V_2(t) = S\left(\frac{180}{\Pi}\right)\tan^{-1}\left(\frac{Q(t)}{I(t)}\right)$$

2. When RefFreq = −1, then internal oscillator frequency synchronization to the input signal is performed as described in *Timed Modem Components* (timed). This synchronization is only allowed when the input signal is an RF (complex envelope) timed signal.
3. When the input signal is a baseband signal, input signal transformation to an RF (complex envelope) timed signal is performed as described in *Demodulation Algorithms* (timed), case 2, before the demodulation process is performed.

PM_DemodExtOsc



Description: Phase demodulator with external reference oscillator
Library: Timed, Modem
Class: TSDF_PM_DemodExtOsc
Derived From: baseOmniSysStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
Sensitivity	voltage output sensitivity, Vout/degree	1		real	(-∞, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed
2	osc	RF oscillator signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
3	output	output signal	timed

Notes/Equations

1. The inphase and quadrature components I(t) and Q(t) of the modulated signal are extracted according to the procedure described in the *Timed Modem Components* (timed). The output at pin 3 is then calculated as:

$$V_3(t) = S\left(\frac{180}{\Pi}\right)\tan^{-1}\left(\frac{Q(t)}{I(t)}\right)$$

2. When the input signal is a baseband signal, input signal transformation to an RF (complex envelope) timed signal is performed as described in *Demodulation Algorithms* (timed), case 2, before the demodulation process is performed.

PM_Mod

Description: Phase modulator with internal oscillator
Library: Timed, Modem
Class: TSDF_PM_Mod
Derived From: baseModulator

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
FCarrier	carrier frequency	1000000	Hz	real	(0, ∞)
Power	modulator output power	0.01	W	real	[0, ∞)
Sensitivity	phase deviation sensitivity, in degrees/volt	1		real	(-∞, ∞)
Phase	phase reference in degrees	0	deg	real	(-∞, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
2	output	output signal	timed

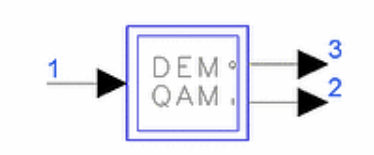
Notes/Equations

1. This component is a phase modulator whose carrier frequency, output power and modulation index can be set by the user.
2. The output of a phase modulator can be expressed as follows:
   $$V_2(t) = A\cos(2\pi f_c t + 2\pi S V_1(t) + 0)$$
   where A is the amplitude of the unmodulated carrier. In the program the signal $V_2(t)$ is represented in terms of its inphase and quadrature components:
   $$V_2(t) = Re\left\{(v_2(t))e^{j\omega_c t}\right\}$$
   $$v_2(t) = v_{I2}(t) + jv_{Q2}(t)$$
   $$v_2(t) = Ae^{j\omega_c t + 2\pi S V_1(t) + \theta}e^{(-j)\omega_c t}$$

$$A = \sqrt{(2)(ROut)10^{\frac{PdBm-30}{10}}}, \quad \omega_c = 2\pi f_c$$

PdBm = Power in dBm units
S = Sensitivity/360
θ = π *Phase* /180

QAM_Demod



Description: Quadrature amplitude demodulator with internal oscillator
Library: Timed, Modem
Class: TSDF_QAM_Demod
Derived From: baseDemodulator

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
RefFreq	internal reference frequency	1000000	Hz	real	{-1} or (0, ∞)†
Sensitivity	voltage output sensitivity, Vout/Vin	1.0		real	(-∞, ∞)
Phase	reference phase in degrees	0.0	deg	real	(-∞, ∞)
GainImbalance	gain imbalance in dB, Q channel relative to I channel	0.0		real	(-∞, ∞)
PhaseImbalance	phase imbalance in degrees, Q channel relative to I channel	0.0	deg	real	(-∞, ∞)

† Refer to Note 2 for the special meaning of a -1 value.

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

Pin Outputs

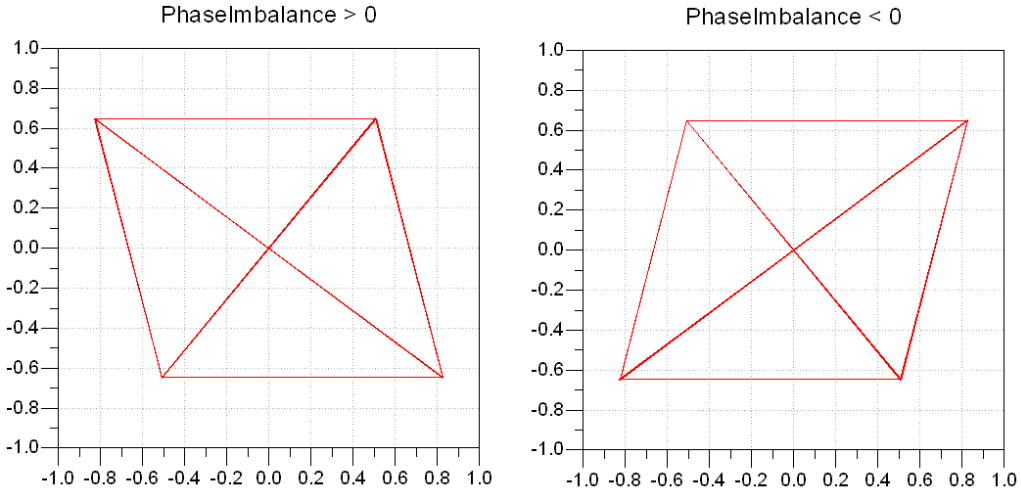
Pin	Name	Description	Signal Type
2	I_out	I output	timed
3	Q_out	Q output	timed

Notes/Equations

1. This demodulator has an internal oscillator that generates the reference carrier signal used to demodulate the RF signal. The inphase and quadrature components I(t) and Q(t) of the modulated signal are extracted as described in the *Timed Modem Components* (timed). Then
$$V_2(t) = S\,I(t) \quad \text{and} \quad V_3(t) = S g Q(t)$$
where S is Sensitivity and

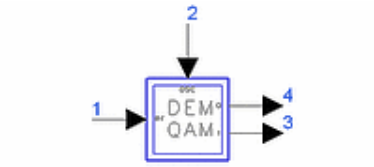
$$g = 10^{\frac{GainImbalance}{20}}$$

2. PhaseImbalance skews the I-axis relative to the ideal I-axis (at 90 degrees with respect to the Q-axis). Positive values for PhaseImbalance cause "clockwise" skews, whereas negative values for PhaseImbalance cause "counter-clockwise" skews.



3. When RefFreq = −1, internal oscillator frequency synchronization to the input signal is performed as described in the *Timed Modem Components* (timed). This synchronization is allowed when the input signal is an RF (complex envelope) timed signal.
4. When the input signal is a baseband signal, input signal transformation to an RF (complex envelope) timed signal is performed as described in *Demodulation Algorithms* (timed), case 2, before the demodulation process is performed.

QAM_DemodExtOsc



Description: Quadrature amplitude demodulator with external reference oscillator
Library: Timed, Modem
Class: TSDF_QAM_DemodExtOsc
Derived From: baseOmniSysStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
Sensitivity	voltage output sensitivity, Vout/Vin	1		real	(-∞, ∞)
GainImbalance	gain imbalance in dB, Q channel relative to I channel	0		real	(-∞, ∞)
PhaseImbalance	phase imbalance in degrees, Q channel relative to I channel	0		real	(-∞, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed
2	osc	RF oscillator signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
3	I_out	I output	timed
4	Q_out	Q output	timed

Notes/Equations

1. This demodulator requires an external reference carrier signal to demodulate the RF signal. The inphase and quadrature components *I (t)* and *Q (t)* of the modulated signals at pins 1 and 2 are extracted according to the procedure described in the *Timed Modem Components* (timed). If the RF signal carrier frequencies at pins 1 and 2 are not the same, the pin 1 I and Q signals are converted to their representation at the pin 2 carrier frequency.
   Let:
   $I_1(t)$ and $Q_1(t)$ represent the pin 1 signal represented at the pin 2 carrier frequency.
   $I_2(t)$ and $Q_2(t)$ represent the pin 2 signal.
   The outputs at pins 3 and 4, $V_3(t)$ and $V_4(t)$, are calculated as follows:

$V_3(t) = S \times (I_1(t) \times I_2(t) + Q_1(t) \times Q_2(t)) / (A2)$

$V_4(t) = S \times AIQ \times (I_1(t) \times VI + Q_1(t) \times VQ) / (A2)$

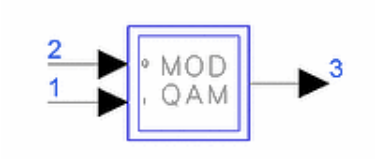
where
S = Sensitivity
$A2$ = magnitude of the signal at pin 2
$VI = I_2(t) \times \cos(PHIQ) - Q_2(t) \times \sin(PHIQ)$

$VQ = I_2(t) \times \sin(PHIQ) + Q_2(t) \times \cos(PHIQ)$

$AIQ = 10^{GainImbalance/20}$
$PHIQ = \pi/2 - PhaseImbalance \times \pi/180$

2. When the input signal is a baseband signal, input signal transformation to an RF (complex envelope) timed signal is performed as described in *Demodulation Algorithms* (timed), case 2, before the demodulation process is performed.

QAM_Mod



Description: Quadrature amplitude modulator with internal oscillator
Library: Timed, Modem
Class: TSDF_QAM_Mod
Derived From: baseModulator

Parameters

Name	Description	Default	Symbol	Unit	Type	Range
RIn	input resistance	DefaultRIn		Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut		Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp		Celsius	real	[-273.15, ∞)
FCarrier	carrier frequency	1000000	f_c	Hz	real	(0, ∞)
Power	modulator output power	0.01	P	W	real	[0, ∞)
VRef	modulator voltage reference level	1		V	real	(0, ∞)
Phase	reference phase in degrees	0	Θ	deg	real	(-∞, ∞)
GainImbalance	gain imbalance in dB, Q channel relative to I channel	0	G		real	(-∞, ∞)
PhaseImbalance	phase imbalance in degrees, Q channel relative to I channel	0	Φ		real	(-∞, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	I_in	I input	timed
2	Q_in	Q input	timed

Pin Outputs

Pin	Name	Description	Signal Type
3	output	output signal	timed

Notes/Equations

1. QAM_Mod takes I and Q baseband input signals and generates a QAM RF signal at its output. For each sample consumed at the two inputs one sample is produced at the output.
2. The output of QAM_Mod can be expressed as

$$V_3(t) = A\left(V_1(t)\cos(\omega_c t + \theta) - gV_2(t)\sin\left(\omega_c t + \theta + \frac{\phi\pi}{180}\right)\right)$$

where A is a constant that depends on ROut, Power, and VRef

$$A = \sqrt{2 \times ROut \times P / VRef}$$

$$\omega_c = 2 \times \pi \times f_c$$

$$g = 10^{G/20}$$

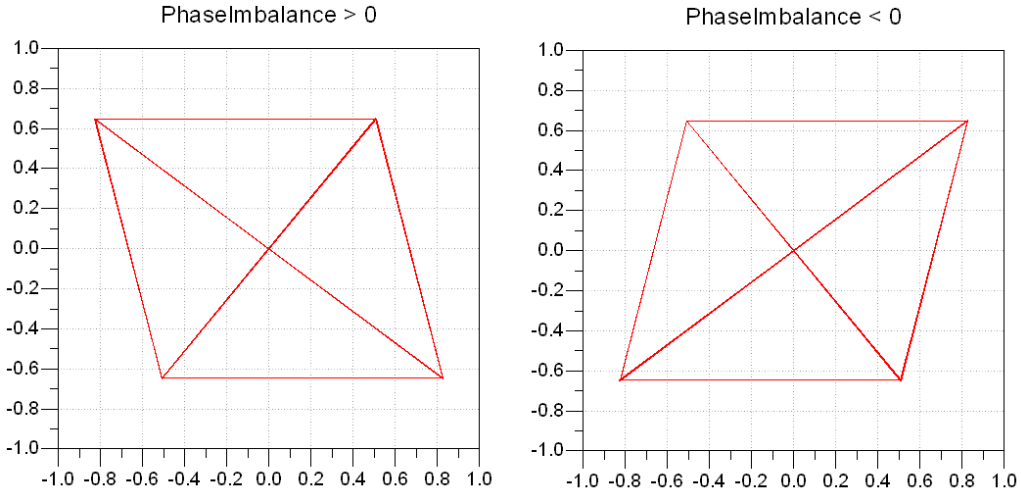
and $V_1(t)$, $V_2(t)$, $V_3(t)$ are the signals at pins 1, 2, 3, respectively. The signal $V_3(t)$ is represented in terms of its inphase and quadrature phase components:

$$V_3(t) = Re\left\{(v_{I3}(t) + jv_{Q3}(t))e^{j(\omega_c \times (t+\theta))}\right\}$$

$$v_{I3}(t) = A\left(V_1(t) - gV_2(t)\sin\left(\frac{\phi\pi}{180}\right)\right)$$

$$v_{Q3}(t) = AgV_2(t)\cos\left(\frac{\phi\pi}{180}\right)$$

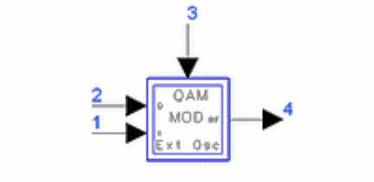
3. PhaseImbalance skews the Q-axis relative to the ideal Q-axis (at 90 degrees with respect to the I-axis). Positive values for PhaseImbalance cause "clockwise" skews, whereas negative values for PhaseImbalance cause "counter-clockwise" skews.



4. In order to get the correct power at the output of QAM_Mod, VRef must be set to the rms value of the input IQ-signal. For example, if the I signal is an NRZ waveform of 1V amplitude and the Q signal is an NRZ waveform of 2V amplitude, then VRef must be set to

$$\sqrt{1^2 + 2^2} = \sqrt{5}$$

QAM_ModExtOsc



Description: Quadrature amplitude modulator with external oscillator
Library: Timed, Modem
Class: TSDFQAM_ModExtOsc
Derived From: baseSubcircuit

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
Power	modulator output power	0.01	W	real	[0, ∞)
VRef	reference voltage for output power calibration	1	V	real	(0, ∞)
GainImbalance	gain imbalance in dB, Q channel relative to I channel	0		real	(-∞, ∞)
PhaseImbalance	phase imbalance in degrees, Q channel relative to I channel	0		real	(-∞, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	I_in	I input	timed
2	Q_in	Q input	timed
3	osc	RF oscillator signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
4	output	output signal	timed

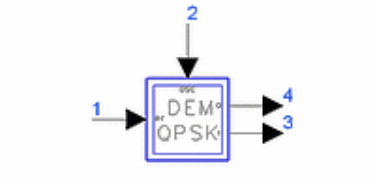
Notes/Equations

1. QAM_ModExtOsc takes as inputs I and Q baseband signals and an RF oscillator signal and generates a QAM RF signal at its output. For each sample consumed at the three inputs one sample is produced at the output.
2. This component is the same as the QAM_Mod component except QAM_ModExtOsc requires an external oscillator signal, whereas QAM_Mod generates the oscillator signal internally. For more details, refer to *QAM_Mod* (timed) documentation.
3. QAM_ModExtOsc is calibrated so that it delivers the correct power on a matched load connected at its output if *VRef* is set to the rms value of the input IQ-signal and the power of the oscillator signal is 10 mW. If the oscillator signal power is not 10 mW, then in order to get the correct power at the component's output, *VRef* must be set

to the input IQ-signal rms value multiplied by *R*, where *R* is the square root of the ratio of the actual oscillator signal power to 10 mW. For example, if the I signal is an NRZ waveform of amplitude 1V, the Q signal is an NRZ waveform of amplitude 2V, and the oscillator signal power is 15 mW, then *VRef* must be set to

$$\sqrt{1^2 + 2^2} \times \sqrt{15/10} = \sqrt{7.5}$$

QPSK_Demod



Description: QPSK demodulator with external reference oscillator
Library: Timed, Modem
Class: TSDFQPSK_Demod
Derived From: baseSubcircuit

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
SymbolTime	input symbol time interval	0.001	sec	real	[TStep, ∞)†
ExcessBw	raised cosine filter excess bandwidth	0	real	[newpro:0, 1]	

† TStep is the simulation time step for the component input signals.

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed
2	osc	RF oscillator signal	timed

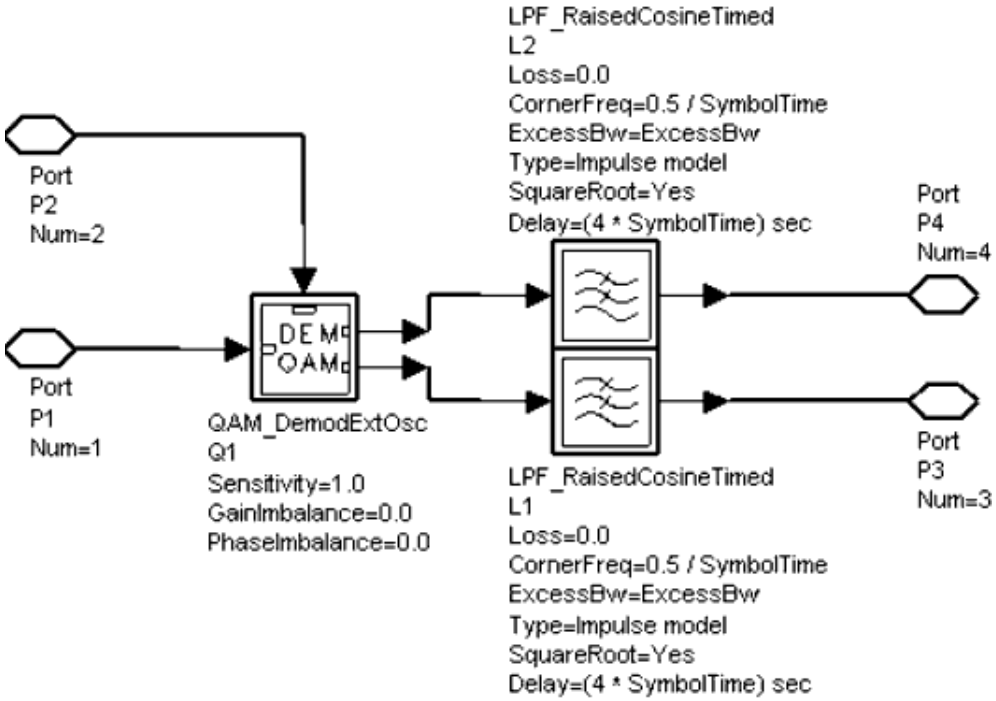
Pin Outputs

Pin	Name	Description	Signal Type
3	I_out	I output	timed
4	Q_out	Q output	timed

Notes/Equations

1. QPSK_Demod is a hierarchical model composed of other components. QPSK Demodulator Network shows the implementation of the network representing the demodulator. The two inputs to the demodulator are the RF signal and a local oscillator signal. The demodulated I and Q signals are filtered with raised-cosine filters.

QPSK Demodulator Network



The following points should be noted.

- The bit time of the data bits is specified by SymbolTime and the rolloff factor of the raised-cosine filters is set by ExcessBw.
- The raised-cosine filters have a corner frequency of 1/(2SymbolTime) Hz.
- The impulse response of the raised-cosine filter has a delay of 4SymbolTime seconds. The QPSK modulator MODQPSK has a similar delay. Therefore, the total delay introduced by the modulator and demodulator is 8SymbolTime seconds. This delay should be increased to 8SymbolTime or greater for greater accuracy, especially when making measurements such as EVM.
- One-half of the raised-cosine filtering is provided in the receiver (by using the square root of the cosine filters) because the remaining half of the filtering is performed at the transmitter.
- RF filtering is not included at the input of this QPSK demodulator; if required, the appropriate filter must be placed by the user.
- Only the phase of the local oscillator signal will affect the output of the demodulator, whereas its amplitude will not affect the output. This is because the QAM_DemodExtOsc component is used for demodulation; and, for this component only the RF signal, the demodulation sensitivity factor S, and the phase of the local oscillator determine the output signal (see the documentation for details). The demodulation sensitivity of the QAM_DemodExtOsc component is set at 1 volt/volt in this network.
- The required input RF power level in order to obtain ±1V at the ideal sampling instances at the output of this demodulator is 13 dBm.

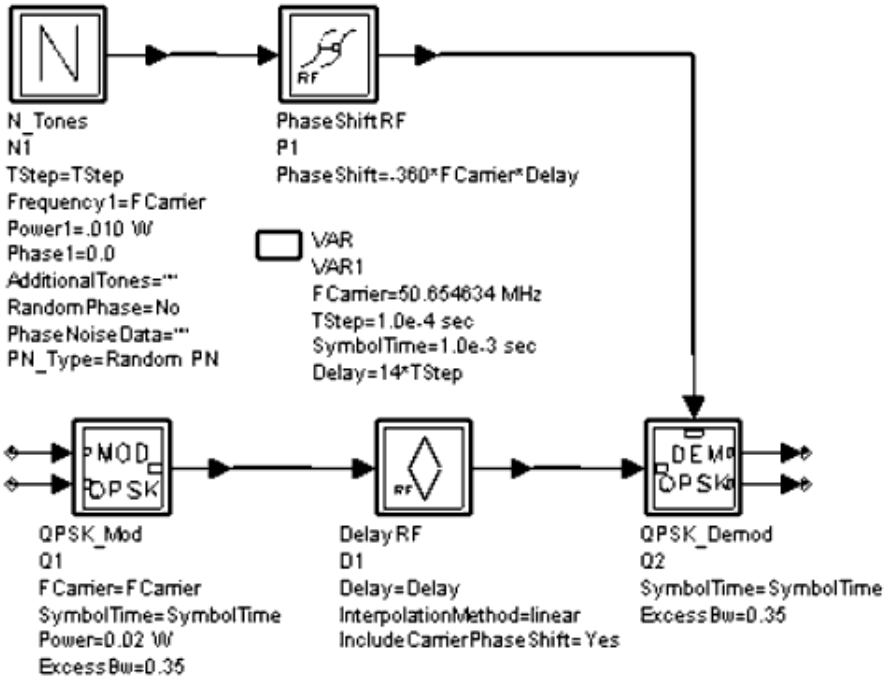
2. When the input signal is a baseband signal, input signal transformation to an RF (complex envelope) timed signal is performed as described in *Demodulation Algorithms* (timed), case 2, before the demodulation process is performed.
3. The phase of the local oscillator signal must be matched to the phase of the received RF signal for proper demodulation of the QPSK signal.
   If the delay from the modulator to the receiver is known, an oscillator component (N_Tones) can be used at the receiver to generate the LO signal and the phase of the oscillator output can be adjusted with the PhaseShiftRF component as shown in QPSK Demodulation with Local Oscillator.

If the channel delay is not known, the QPSK_Recovery component can be used to recover the carrier from the RF signal. The recovered carrier from the QPSK_Recovery component has a phase ambiguity of 90 degrees and multiples thereof. Therefore, differential encoding and decoding is required with this carrier recovery scheme.

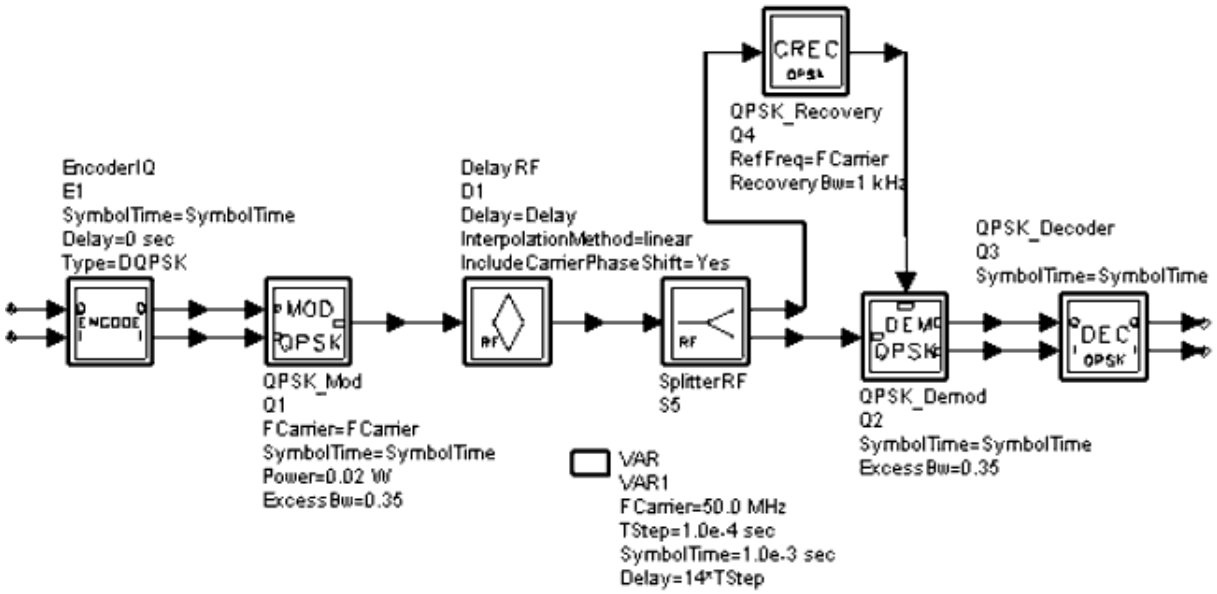
Differential encoding can be done by the EncoderIQ component with Type=DQPSK; differential decoding can be done with the QPSK_Decoder component. (With differential coding and decoding, the first pair of decoded bits may be in error because the starting state of the differential coder is not known by the differential decoder.) QPSK Demodulation with Carrier Recovery illustrates how such a system can be set up.

4. The BER performance of the QPSK_Mod/QPSK_Demod pair for a distortionless additive white Gaussian channel is shown in BER Performance of QPSK. This assumes ideal carrier and bit timing recovery.

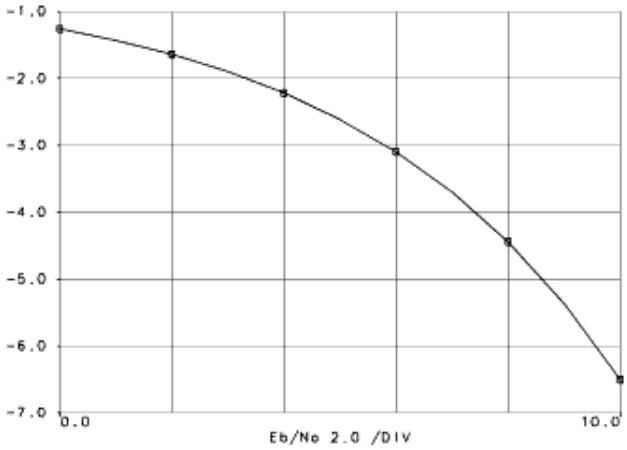
QPSK Demodulation with Local Oscillator



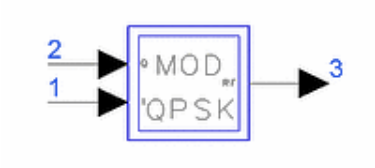
QPSK Demodulation with Carrier Recovery



BER Performance of QPSK



QPSK_Mod



Description: QPSK modulator with internal oscillator
Library: Timed, Modem
Class: TSDFQPSK_Mod
Derived From: baseSubcircuit

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
FCarrier	carrier frequency	1000000	Hz	real	(0, ∞)
SymbolTime	input symbol time interval	0.001	sec	real	[TStep, ∞)†
Power	modulator output power	0.01	W	real	[0, ∞)
ExcessBw	raised cosine filter excess bandwidth	0	real	[0, 1]	

† TStep is the simulation time step for the component input signals.

Pin Inputs

Pin	Name	Description	Signal Type
1	I_in	I input	timed
2	Q_in	Q input	timed

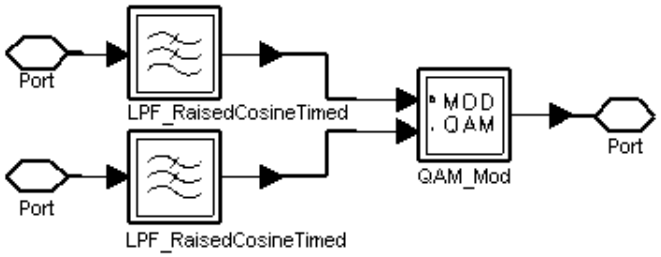
Pin Outputs

Pin	Name	Description	Signal Type
3	output	output signal	timed

Notes/Equations

1. This component is a hierarchical model composed of other components. QPSK Modulator Schematic shows the implementation of the network representing the modulator. The two inputs to the modulator are the I and Q data streams in NRZ format. The NRZ data symbols are filtered with root-raised-cosine filters then modulated to the carrier frequency with a QAM modulator.

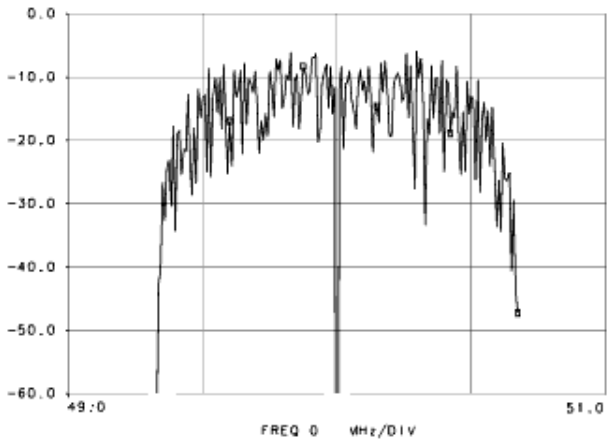
QPSK Modulator Schematic



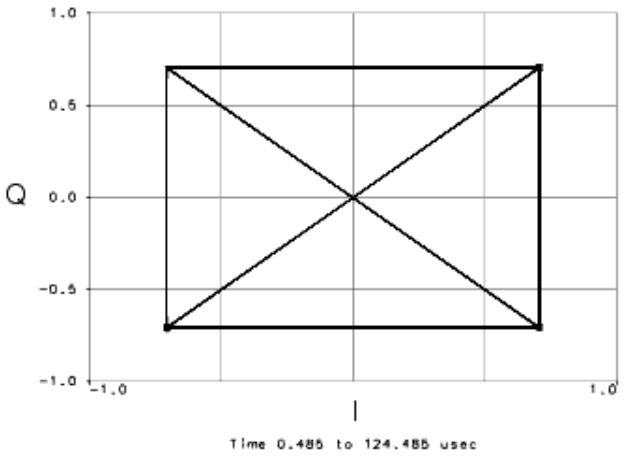
The following points should be noted about the transmitter:

- The bit period of the input data is specified by SymbolTime; the roll-off factor of the raised-cosine filters is set by ExcessBw; and, the carrier frequency and power of the output signal are set by FCarrier and Power, respectively. In order to get the correct power at the output of the component, both the I and Q input NRZ waveforms must have a 1V amplitude.
- Only half of the raised-cosine filtering is provided in the transmitter (by using the square root of the cosine filters); the remaining half of the filtering should be performed at the receiver.
- The raised-cosine filters have an (f/sin(f)) shaped equalization to account for the finite width of the input NRZ data.
- The raised-cosine filters have a corner frequency of 1/(2SymbolTime) Hz.
- The impulse response of the filters has a delay of 4SymbolTime seconds. This delay should be increased to 8SymbolTime or greater for better accuracy, especially when making measurements such as EVM.
  The spectrum of the modulated signal is shown in Output Spectrum (Power (dBm) versus Freq) of QPSK Modulator for the following values:
  FCarrier = 50 MHz
  SymbolTime = 1 µsec
  Power = 10 dBm
  ExcessBw = 0.35
  The IQ constellation diagram of the modulated signal, after filtering with a bandpass square root raised-cosine filter, is shown in the QPSK Constellation Diagram.

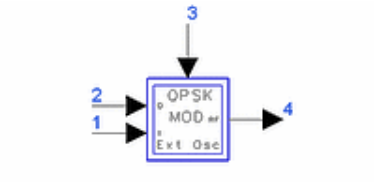
Output Spectrum (Power (dBm) versus Freq) of QPSK Modulator



QPSK Constellation Diagram


1.0
0.5
0.0
-0.5
-1.0
Q
-1.0
1.0
I
Time 0.485 to 124.485 usec

QPSK_ModExtOsc



Description: QPSK modulator with external oscillator
Library: Timed, Modem
Class: TSDFQPSK_ModExtOsc
Derived From: baseSubcircuit

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
SymbolTime	input symbol time interval	0.001	sec	real	[TStep, ∞)†
Power	modulator output power	0.01	W	real	[0, ∞)
ExcessBw	raised cosine filter excess bandwidth	0		real	[newpro:0, 1]

† TStep is the simulation time step for the component input signals.

Pin Inputs

Pin	Name	Description	Signal Type
1	I_in	I input	timed
2	Q_in	Q input	timed
3	osc	RF oscillator signal	timed

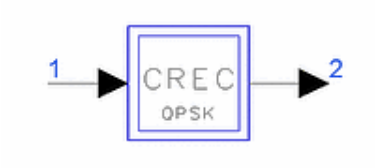
Pin Outputs

Pin	Name	Description	Signal Type
4	output	output signal	timed

Notes/Equations

1. This component is the same as the QPSK_Mod component except QPSK_ModExtOsc requires an external oscillator signal, whereas QPSK_Mod generates the oscillator signal internally. For more details, refer to *QPSK_Mod* (timed) documentation.
2. QPSK_ModExtOsc is calibrated so that it delivers the correct power on a matched load connected at its output if both the I and Q input NRZ waveforms have a 1V amplitude and the power of the oscillator signal is 10 mW.

QPSK_Recovery



Description: QPSK carrier recovery
Library: Timed, Modem
Class: TSDFQPSK_Recovery
Derived From: baseSubcircuit

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
RefFreq	internal reference frequency	1000000	Hz	real	(0, ∞)
RecoveryBw	bandwidth of carrier recovery filter	1000	Hz	real	(0, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

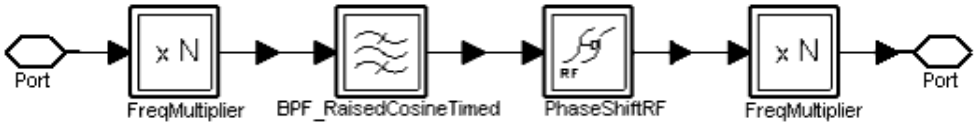
Pin Outputs

Pin	Name	Description	Signal Type
2	output	output signal	timed

Notes/Equations

1. This component is composed of other components. QPSK_Recovery Carrier Recovery Network shows the implementation of the network representing the QPSK carrier recovery circuit.
   The carrier is recovered from the RF signal by taking the fourth power of the input signal, extracting the resulting tone at 4RefFreq, and then dividing the filtered output by 4 to obtain the carrier.
   The filter is an ideal bandpass filter centered at a frequency of 4RefFreq with a bandwidth of RecoveryBw. The filter has a delay of 4/RecoveryBw associated with it. The PhaseShiftRF component compensates for the phase shift introduced by this delay.
   The spectral purity of the recovered carrier increases as the bandwidth of the filter is decreased-however, the delay through the circuit increases proportionally.

QPSK_Recovery Carrier Recovery Network


Port
x N
FreqMultiplier
BPF_RaisedCosineTimed
RF
PhaseShiftRF
x N
FreqMultiplier
Port

Timed Nonlinear Components

The Timed Nonlinear component library contains components for nonlinear gain blocks, RF signal mixers, RF frequency converters, and more. All nonlinearities are modeled as bandpass zero memory nonlinearities. For timed baseband signals, the simulation (single sided) lowpass bandwidth is equal to 1/2/TStep. For timed RF (complex envelope) signals, the simulation bandpass bandwidth is equal to 1/TStep. The RF output from a nonlinear process includes only those RF intermodulation products that appear in this first zone about the RF carrier; all other intermodulation products (odd or even) are ignored. This bandpass characteristic of nonlinear processes is a commonly accepted assumption for RF communication signals. The typical application assumes the information bandwidth is much smaller than the RF carrier frequency. This bandpass assumption is not valid for wideband RF signals that must include harmonics of the RF carrier. Applications with wideband RF signals that include RF harmonics can be simulated using the Circuit Envelope simulator available in ADS circuit schematics.

Each timed nonlinear component consumes and produces baseband or RF (complex envelope) timed signals. If a component receives another class of scalar signal, the received signal is automatically converted to a timed type of signal. Auto conversion from the complex scalar to the timed class is not allowed. The user must explicitly use the CxToTimed signal converter for this transformation. These components do not accept any matrix class of signal.

Some components will only accept or produce a baseband or RF (complex envelope) timed signal. If a component requires an RF (complex envelope) timed signal, then its receipt of a baseband timed signal will be declared an error and the simulation will stop. If a baseband timed signal is the required input, then any received RF (complex envelope) timed signal is first transformed into its baseband equivalent form before use by the component.

An RF (complex envelope) timed signal is converted to its equivalent baseband form:

$$V_{bb}(t) = Re\left\{(v_{RF}(t))e^{j2\pi f_c t}\right\} = Re\left\{(v_I(t) + jv_Q(t))e^{j2\pi f_c t}\right\}$$

where
$V_{bb}(t)$ is the total representation for the RF signal (also called the baseband representation)
$v_{RF}(t)$ is the RF signal complex envelope at characterization frequency fc (also called the equivalent complex baseband envelope representation for the RF signal)
$v_I(t)$ is the RF timed signal in-phase envelope
$v_Q(t)$ is the RF timed signal quadrature-phase envelope
f_c is the RF signal characterization frequency

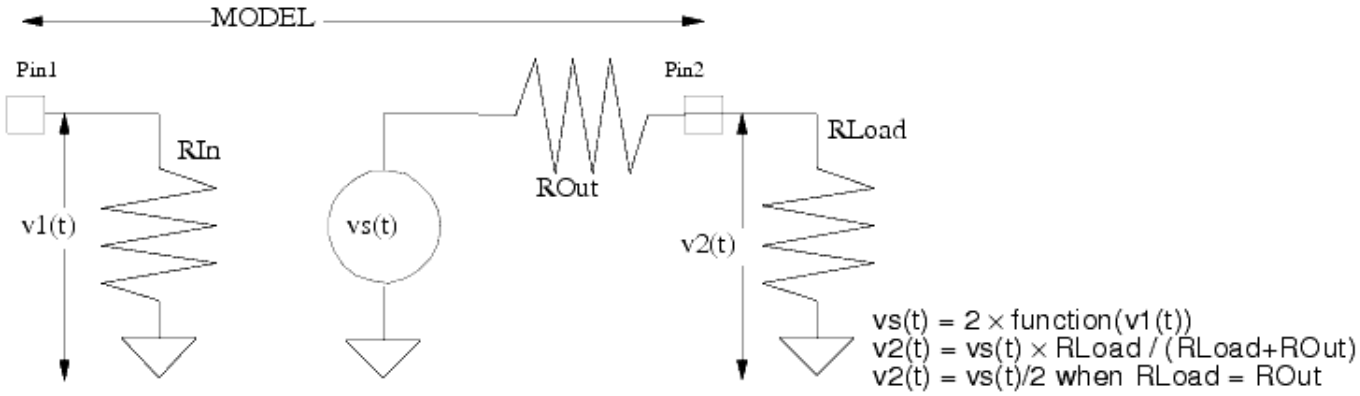
For this equivalence to be valid, the simulation time step must be less than the inverse of the characterization frequency and the RF signal information content has an information bandwidth less than the RF characterization frequency.

All nonlinear components have user-specified input (RIn) and output (ROut) resistance. Input resistance is for a shunt resistor at each component input pin; output resistance is

for a series resistor at each component output pin.

Each component internal output signal vs(t), has a value equal to twice the output pin signal v2(t), when the output is connected to a matched resistor load. The circuit model illustrated in 2-Port Circuit Model for a 2-port (1 input, 1 output) timed nonlinear component demonstrates this relationship.

2-Port Circuit Model



Note:
A scale factor of 2 is used in the vs(t) expression so that when ROut=RLoad the voltage across RLoad will be exactly function(vs(t)).

The output pin signal, v2(t), at the output series resistance is dependent on the value of the load resistance connected to it. When the load resistor RLoad is equal to the model output resistor ROut the value of v2(t) is equal to vs(t)/2, otherwise, based on the voltage divider action v2(t) is:
$v\,2\,(t) = vs(t) \times RLoad/(RLoad + ROut\ )$

The input and output resistor values must be greater than 0 ohm.

The output resistors contribute additive thermal noise power (kTB) to the output signal when the specified resistance temperature (RTemp) is greater than absolute zero (−273.15 °C).

When noise figure (NoiseFigure) is a parameter for a nonlinear component, it contributes additive Gaussian noise power ($k \times T_o \times G \times (F-1) \times B$) to the output signal.

For the above relationships, the following are defined:
k = Boltzmann's constant
T = temperature in Kelvin
T_o = standard reference temperature, 290 Kelvin
G = component gain
F = component noise factor = 10 ^(NoiseFigure/10)
B = simulation frequency bandwidth:

- 1/2/tstep if signal is a timed baseband signal
- 1/tstep if signal is a timed complex envelope signal

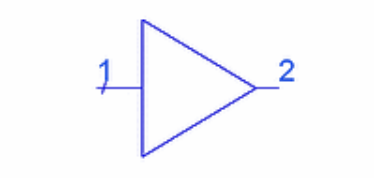
When RTemp > −273.15 or NoiseFigure > 0, the noise contributed from each resistor or noise figure instance is an independent noise process. This noise is dependent on the value of DefaultSeed in the DF (data flow) controller. When DefaultSeed=0, the noise generated for each simulation is different; when DefaultSeed>0, the noise generated for each simulation, though random, has the same initial seed starting condition and thus results in reproducible noise for each simulation.

Note
Information regarding time domain signal differences between ADS Ptolemy simulations and Circuit Envelope and Transient simulations is given in the *Timed Synchronous Dataflow* (ptolemy) section in the *ADS Ptolemy Simulation* (ptolemy) documentation.

Components

- *CktAmp* (timed)
- *FreqMultiplier* (timed)
- *GainRF* (timed)
- *Limiter* (timed)
- *LogAmp* (timed)
- *LogVDet* (timed)
- *MixerRF* (timed)
- *MultiplierRF* (timed)
- *PeakDetector* (timed)
- *PhaseCompRF* (timed)
- *PhaseFreqComp* (timed)
- *PhaseFreqCompXOR* (timed)
- *RateLimiter* (timed)
- *Rectifier* (timed)
- *TimedSDC1* (timed)
- *TimedSDC2* (timed)
- *TimedSDC3* (timed)
- *TimedSDC4* (timed)
- *VcGainRF* (timed)

CktAmp



Description: Circuit behavioral nonlinear amplifier model
Library: Timed, RF Subsystems
Class: TSDFCktAmp

Parameters

Name	Description	Default	Unit	Type	Range
TStep	Circuit envelope simulation time step	1 nsec	sec	real	(0, +∞)
FCarrier	Circuit envelope simulation carrier frequency	1 GHz	Hz	real	(0, +∞)
Gain	Complex gain	1		complex	
NF	dB noise figure	0	dB	real	[0, +∞)
dBc1out	Output power at 1dB gain compression	0.01	W	real	[0, +∞)
Delay	Output delay	0	sec	real	{0} or [TStep, 50*TStep]
MirrorSignal	Mirror Signal? NO, YES	NO		enum	

Pin Inputs

Pin	Name	Description	Signal Type
1	RF_in	RF input signal	timed

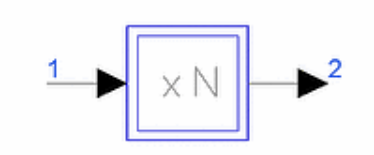
Pin Outputs

Pin	Name	Description	Signal Type
2	RF_out	RF output signal	timed

Notes/Equations

1. CktAmp is a circuit behavioral amplifier model that uses Circuit Envelope analysis.
2. TStep is used to set the Circuit Envelope analysis time step. The Circuit Envelope analysis time stop is set to 100 × TStep.
3. FCarrier is used to set the Circuit Envelope analysis frequency. This is the only frequency used in Circuit Envelope analysis. The order of Circuit Envelope analysis is set to 1.
4. dBc1out sets the output power at the 1dB gain compression point.

FreqMultiplier



Description: Signal frequency multiplier
Library: Timed, Nonlinear
Class: TSDF_FreqMultiplier
Derived From: baseStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
X	frequency multiplication factor	1		real	(0, ∞)
Type	multiplier type: full signal, RF phase only	full signal		enum	

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
2	output	output signal	timed

Notes/Equations

1. FreqMultiplier is a computationally efficient model of a frequency multiplier. Frequency multiplication is accomplished by first passing the input signal through a nonlinearity (which raises the input signal to the power n), then passing the signal through a bandpass filter centered at n × f_{c1} , where f_{c1} is the carrier frequency of the input signal. Therefore, this model accepts only RF signals at its input. Computational efficiency is accomplished by directly calculating the output of the bandpass filter without explicitly performing the filtering operation.
   Assuming the input signal is

$$V_1(t) = Re\left\{(v_{I1}(t) + j \times v_{Q1}(t)) \times e^{j2\pi f_{c1} t}\right\}$$

   the output signal is given by

$$V_2(t) = Re\left\{v_{2mag} \times (\cos(v_{2ph}) + j \times \sin(v_{2ph})) \times e^{j2\pi X f_{c1} t}\right\}$$

   where

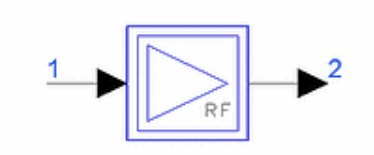
$$v_{2mag} = (\sqrt{v_{I1} \times v_{I1} + v_{Q1} \times v_{Q1}})^{a}$$

a=1 when Type=RF phase only
a=X when Type=full signal
and

$$v_{ph} = X \times \text{atan2}(v_{Q1}, v_{I1})$$

2. Frequency multiplication is accomplished by setting X > 1 while frequency division is obtained by setting 0 < X < 1.
   In the case when 0 < X < 1 (the component is used as a frequency divider), the user must ensure that the output signal is a bandpass signal (that is, the signal does not have any significant energy at 0 Hz). This is because the output is represented in terms of its complex envelope, assuming that the signal is indeed a bandpass signal. If this condition is not met, the user must convert the complex envelope representation to a baseband representation by placing an FcChange component (with FC=0) at the output of FreqMultiplier.
3. For information regarding timed nonlinear component signals, refer to *Timed Nonlinear Components* (timed).

GainRF



Description: Complex gain with gain compression
Library: Timed, Nonlinear
Class: TSDF_GainRF
Derived From: baseNoiseFigureStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
Gain	Complex voltage gain; with form re+j*im; to specify gain in dB use dbpolar(dB, degree)	1		complex	(-∞, ∞)
NoiseFigure	input noise figure, in dB	0		real	[0, ∞)
GCType	Gain compression type: none, TOI, dBc1, TOI+dBc1, PSat+GCSat+TOI, PSat+GCSat+dBc1, PSat+GCSat+TOI+dBc1, Gain compression data points vs input power	none		enum	
TOIout	Third order intercept power	3	W	real	(-∞, ∞)†
dBc1out	1 dB gain compression power	1	W	real	(-∞, ∞)†
PSat	Saturation power	1	W	real	(-∞, ∞)†
GCSat	Gain compression at saturation; dB	1		real	[3, 7]†
GComp	Array of triple values for large signal gain change vs signal power. Input Power in dBm, Gain change from small signal in dB, and Phase change from small signal in degree	0 0 0		real array	††

† Values for TOIout, dbc1out, PSat, and GCSat are interdependant as explained in Note 3.
†† Refer to the Note 3 section <i>Gain compression data points vs input power.

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
2	output	output signal	timed

Notes/Equations

1. GainRF can be used to model an amplifier with nonlinear gain compression.
2. The following equations describe the algorithm used for this model.
   The input signal $V_1(t)$ at pin 1 is represented by its inphase and quadrature components about its carrier frequency.

$$V_1(t) = Re\left\{v_1(t)e^{j2\pi f_{c1}t}\right\}, v_{1(t)} = v_{I1}(t) + jv_{Q1}(t)$$

   The output signal $V_2(t)$ at pin 2 is then given by the equation

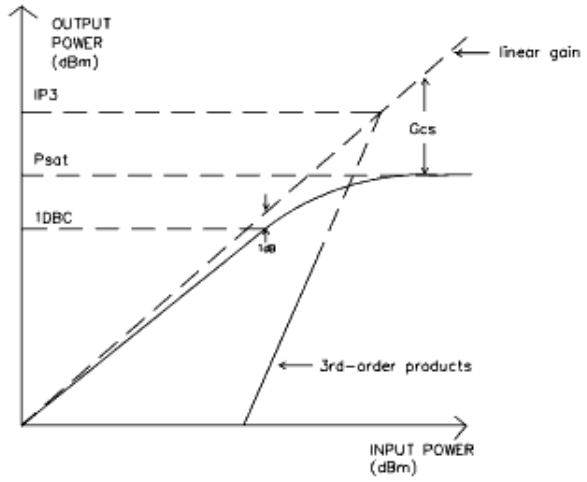
$$V_2(t) = Re\left\{a\, g_{comp} v_1(t) e^{j2\pi f_{c1}t}\right\}$$

   where
   a denotes the gain of the component as set by component parameter Gain. If the input is a baseband timed signal, then only the real part of the Gain is used for a.
   g_{comp} denotes the gain compression factor as determined by the gain compression parameters, GCType, TOIout, dBc1out, PSat, GCSat, GComp
   If GCType≠Gain compression data points vs input power or $f_{c1} = 0.0$, then the imaginary part of g_{comp} is zero.

   If GCType=none, then g_{comp} is one.

   Nonlinear Element Characterization for Nonlinear Models 1 through 6 shows characterization parameters for nonlinear elements.
   All gain compression characteristics (except for GCType= *Gain compression data points vs input power)* are modeled using a polynomial expression up to the saturation point; after this point, output power is held constant for increasing input power. The gain compression expression for nonlinear models is defined with a nonlinear amplitude characteristic.
   For GCType= *Gain compression data points vs input power* the gain compression characteristic is modeled with linear interpolation between the nonlinear data points.

Nonlinear Element Characterization for Nonlinear Models 1 through 6



Nonlinear models *TOI* through *PSat+GCSat+TOI+dBc1* mathematical gain model:

$$V_2(V_1) = a_1 V_1 + a_3 V_1^3 + a_5 V_1^5 + \ldots$$

where

V_1 = input signal voltage

V_2 = output signal voltage

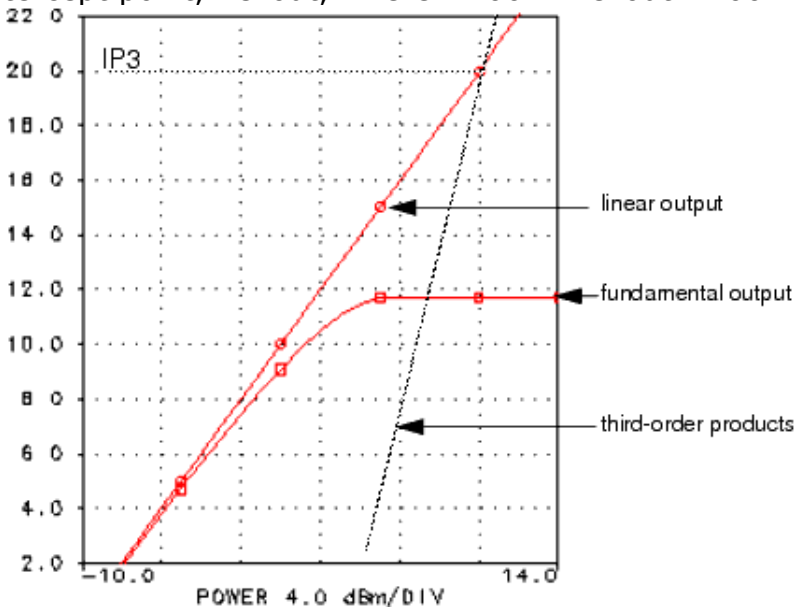
a_1 = small signal gain

a_3 = third-order gain coefficient

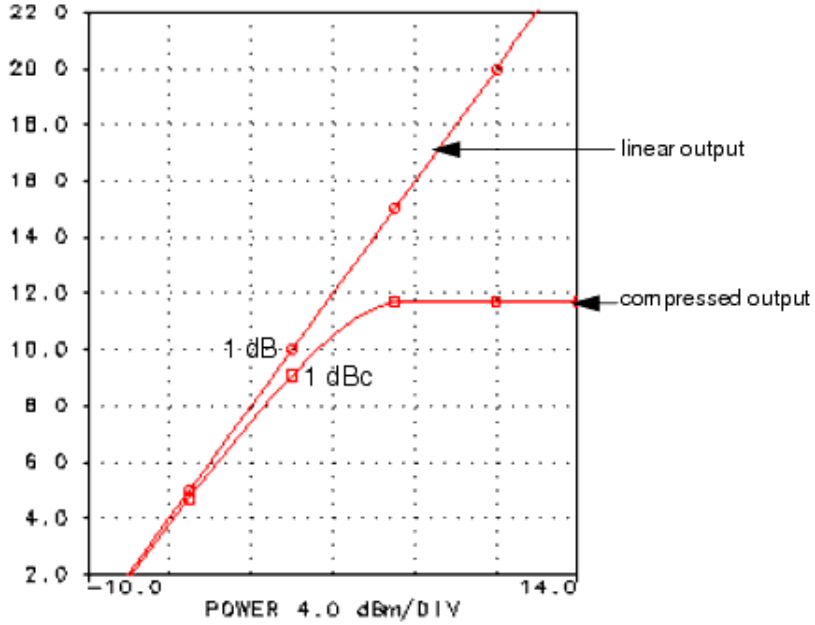
a_5, ... = higher odd-order gain coefficients

The gain compression expression for nonlinear model *Gain compression data points vs input power* may, in general, have both amplitude and phase changes versus increasing input power, as determined by the data you specify.

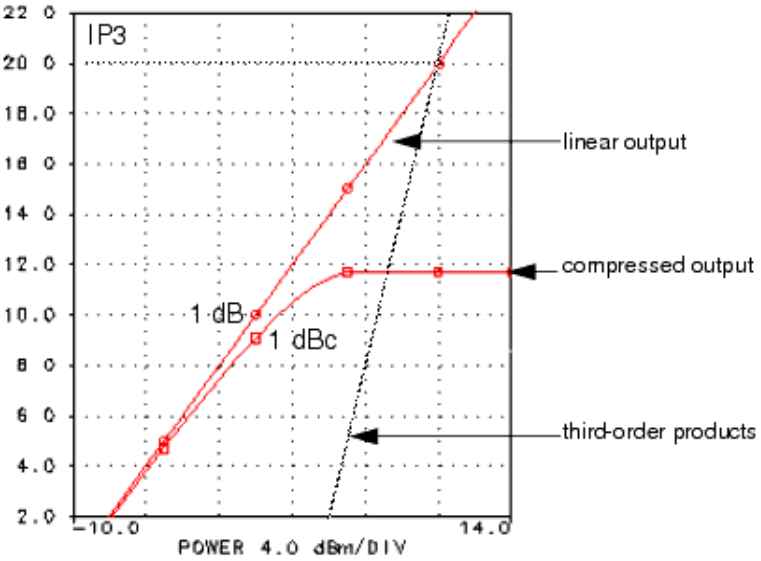
3. When GCType = *TOI*, then the g_{comp} factor is due only to the output third-order intercept point, TOIout, where −100 ≤ TOIout ≤100.



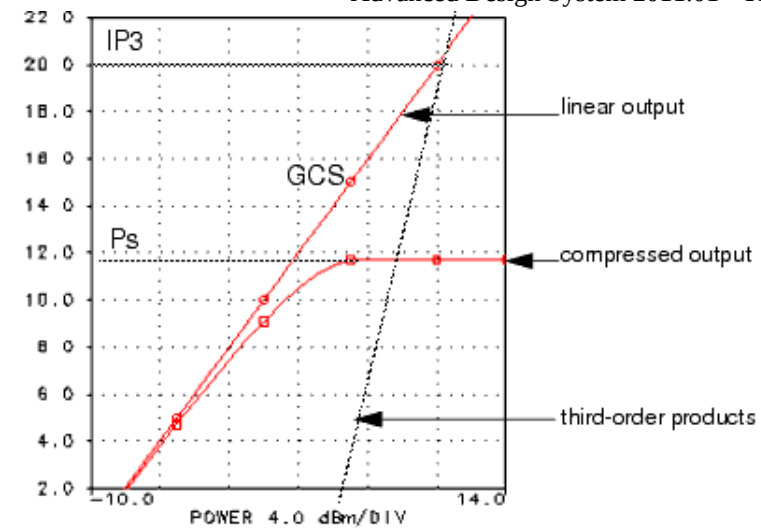
When GCType = *dBc1*, then the g_{comp} factor is due only to the output 1 dB gain compression point, dBc1out, where −100 ≤ dBc1out ≤100.



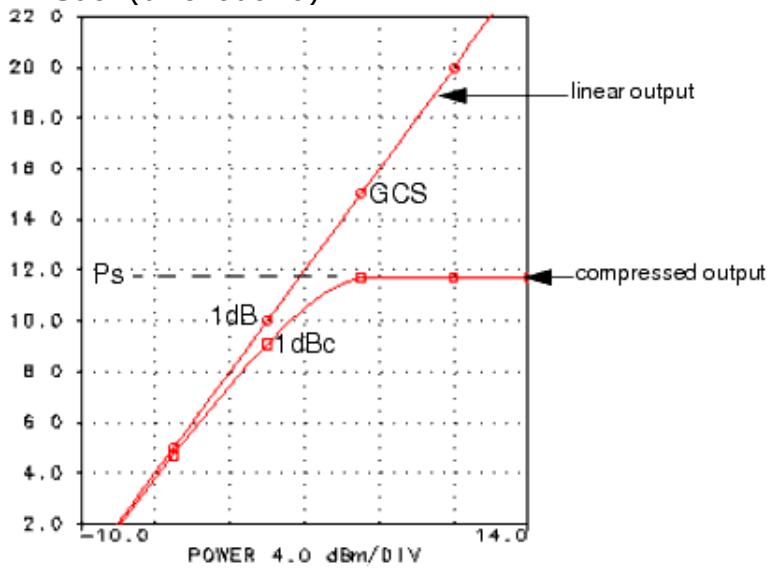
When GCType = *TOI+dBc1*, then the g_{comp} factor is due only to the output third-order intercept point, TOIout, and output 1 dB gain compression point, dBc1out, where (TOI−13.6) ≤ dBc1out ≤(TOI−4.6).



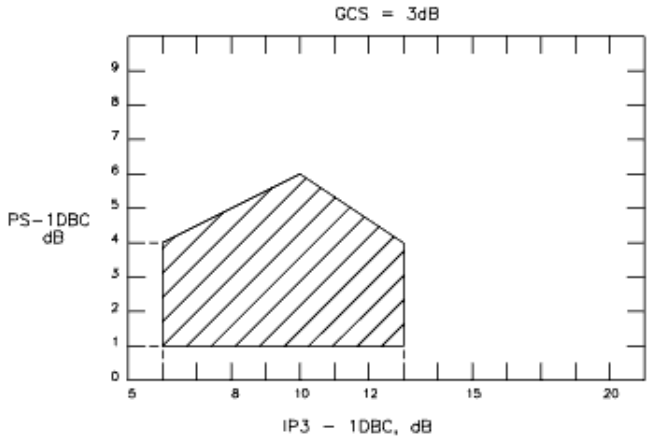
When GCType = *PSat+GCSat+TOI*, then the g_{comp} factor is due only to the output third-order intercept point TOIout, output saturated power PSat, and the gain compression at saturation GCSat, where (3≤GCSat≤7, and (TOI−10+0.5(GCSat-1)) ≤ PSat≤(TOI−4).

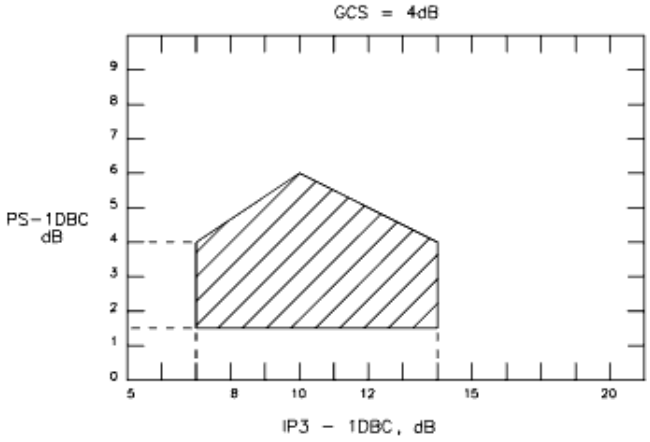


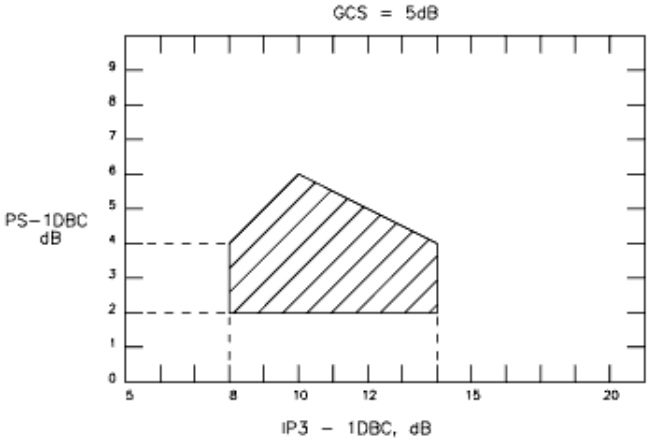
When GCType = *PSat+GCSat+dBc1*, then the g_{comp} factor is due only to the output 1dB gain compression point dBc1out, output saturated power PSat, and the gain compression at saturation GCSat, where (3≤GCSat≤7, and (dBc1out+0.5(GCSat−1)) ≤ PSat≤(dBc1out+6).

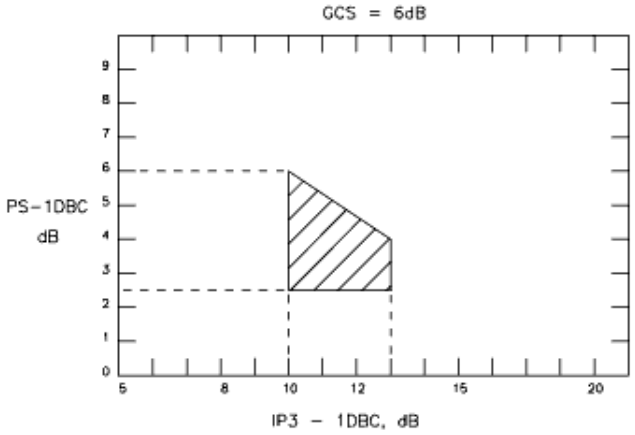


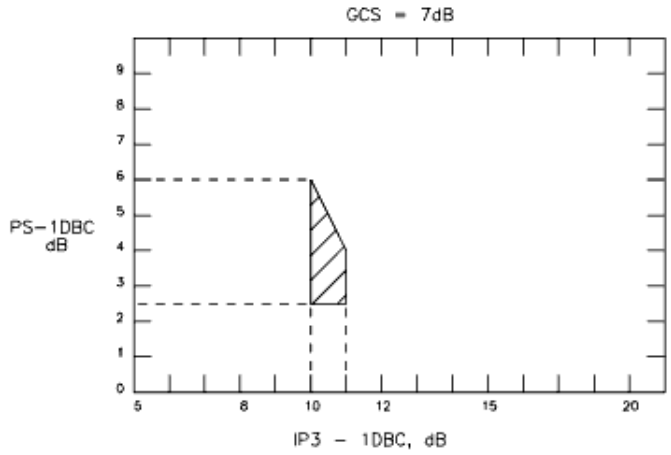
When GCType = *PSat+GCSat+TOI+dBc1*, then the g_{comp} factor is due only to the output third-order intercept point TOI, 1dB gain compression point dBc1out, output saturated power PSat, and the gain compression at saturation GCSat, where the range for valid parameter values is shown in the shaded areas in the following figures:

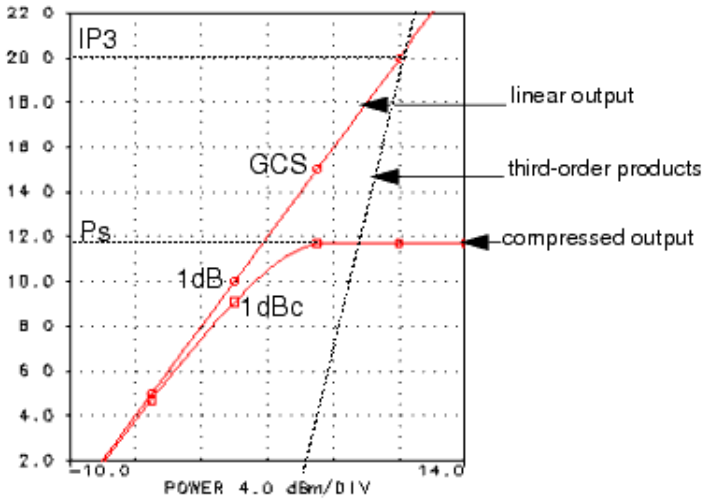




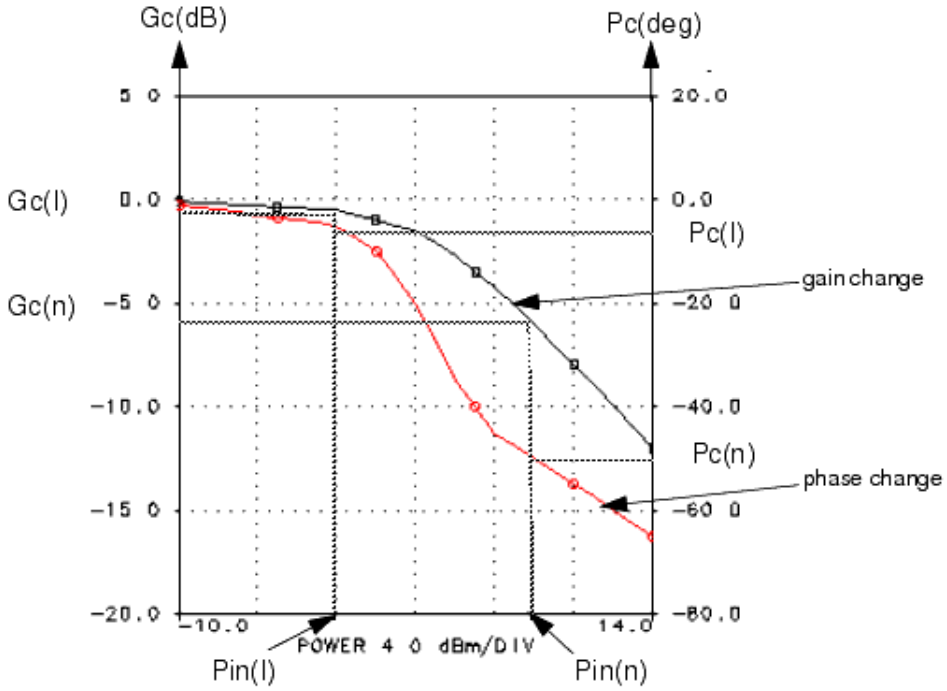








When GCType = *Gain compression data points vs input power*, then the g_{comp} factor is due only to the data in the GComp array. This nonlinear model has a dataset that defines input power (in dBm) and gain change, amplitude change (in dB) and phase change (in degrees), from the small signal gain. Gain and phase change between the given points is found using linear interpolation.



The GComp data points are defined as an array. This array contains triple values of input power (dBm), gain change from small signal (dB), and phase change from small signal (degrees).

```
GComp = "-60 -0.1 0 -50 -0.3 +5 -40 -0.5 -6 -30 -1 8 -20 -2 -6"
```

As an alternative to listing these data points, this dataset can be contained in a text file and referenced by name:

```
GComp = "<mygcomp_data.re"
```

Where the *mygcomp_data.re* file must be located in the current workspace data subdirectory. If not in the subdirectory, then the file name must include the full directory path as the prefix to the filename. The contents of this file simply triple values for each power point where the number separator can be a comma, space, tab, or new line:

-60 -0.1 0
-50 -0.3 +5
-40 -0.5 -6
-30 -1 8
-20 -2 -6

For more details on how to set array parameters see *Array Parameters* (ptolemy).

4. For details on complex parameter values, refer to *Complex-Valued Parameters* (ptolemy) in the *ADS Ptolemy Simulation* (ptolemy) documentation.
   For details on using complex arrays of data, refer to *Value Types* (ptolemy) in the *ADS Ptolemy Simulation* (ptolemy) documentation.
5. Noise may be contributed to the output signal from output resistor ROut and noise figure NoiseFigure. For no noise contribution from the resistor, set its temperature to (or below) absolute zero (−273.15). For no noise contribution from the noise figure, set its value to zero.
   Assuming matched source, load, and component input/output resistor conditions are all 50 ohms, the output noise power is given by

   $$N_{out} = N_{in} \times G_p + k \times T_e \times B \times G_p + k \times T_r \times B$$

   where

   N_{out} = output noise power

   N_{in} = input noise power

G_p = amplifier power gain (=Gain ^{2} since Gain is a voltage gain)

k = Boltzmann's constant = 1.38066e-23

T_e = equivalent noise temperature due to NoiseFigure = (F-1) × T_o

T_o = 290 Kelvin

$F = 10^{NoiseFigure/10}$

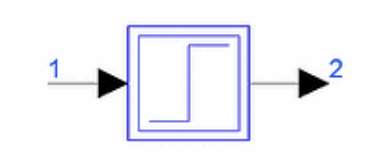
B = simulation bandwidth; 1/TStep for RF signals; 1/2/TStep for baseband signals

T_r = output resistor equivalent noise temperature = RTemp + 273.15

Appropriate scale factors apply for other source, load, and component input/output resistor conditions.

6. For information regarding timed nonlinear component signals, refer to *Timed Nonlinear Components* (timed).

Limiter



Description: Signal limiter
Library: Timed, Nonlinear
Class: TSDF_Limiter
Derived From: baseOmniSysStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
NLimit	negative output saturation voltage, forced equal to -PLimit if input FCarrier > 0	-1	V	real	(-∞, 0)†
PLimit	positive output saturation voltage	1	V	real	(0, ∞)
Gain	non-saturated gain	1		real	(-∞, ∞)

† NLimit defaults to -PLimit when input is a timed RF (complex envelope) signal.

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
2	output	output signal	timed

Notes/Equations

1. Limiter models a soft limiter with linear gain factor and positive and negative saturation levels that can be set by the user. If the input signal is an RF (complex envelope) signal, then the listed value for NLimit is ignored and set equal to −PLimit.
2. Two different models are used for this soft limiter component, depending on whether the input is a baseband or an RF signal (that is, a signal represented by its complex envelope).
   Case 1: When the input is a baseband signal, that is:

$$V_1(t) = Re\left\{(v_{I1}(t) + jv_{Q1}(t))e^{j2\pi f_{c1}t}\right\}$$

$$v_{Q1}(t) = f_{c1} = 0$$

then

$$V_2(t) = Re\left\{(v_{I2}(t) + jv_{Q2}(t))e^{j2\pi f_{c2}t}\right\}$$

where

$$v_{I2}(t) = \begin{cases} NLimit, & \text{if } Gain \cdot v_{I1}(t) < NLimit \\ Gain \cdot v_{I1}(t), & \text{if } NLimit \le Gain \cdot v_{I1}(t) \le PLimit \\ PLimit, & \text{if } PLimit < Gain \cdot v_{I1}(t) \end{cases}$$

$$v_{Q2}(t) = f_{c2} = 0$$

Case 2: When the input is an RF signal, that is, $f_{C1} > 0$

$$V_2(t) = \begin{cases} Gain \cdot V_1(t), & \text{if } Gain \cdot |V_1(t)| \le PLimit \\ \dfrac{PLimit \cdot V_1(t)}{|V_1(t)|}, & \text{if } Gain \cdot |V_1(t)| > PLimit \end{cases}$$

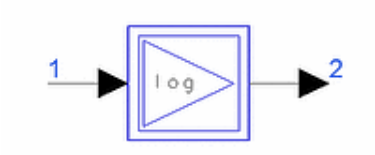
where

$$|V_1(t)| = \sqrt{v_{I1}(t)^2 + v_{Q1}(t)^2}$$

is the magnitude of the signal.

3. For information regarding timed nonlinear component signals, refer to *Timed Nonlinear Components* (timed).

LogAmp



Description: Logarithmic amplifier
Library: Timed, Nonlinear
Class: TSDF_LogAmp
Derived From: baseOmniSysStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
Sensitivity	log sensitivity, in volts/dB	0	V	real	(0, ∞)
PMin	minimum input power, in dBm, for logarithmic amplification	-80		real	(-∞, ∞)
E	peak log error, in dB	0		real	[0, ∞)
Ec	log error cycle, in dB	0		real	(0, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
2	output	output signal	timed

Notes/Equations

1. LogAmp can be used to model a logarithmic amplifier with some nonideal behavior. The basic equation describing such an amplifier is:
   $$V_2(t) = K\log(V_1(t)/V_L) + \varepsilon$$
   where K determines the logarithmic slope, V_L determines the minimum input voltage required for logarithmic amplification, and ε is the deviation of the amplifier from the ideal.
2. The following equations describe the algorithm used for this model.
   Let
   $$V_1(t) = Re\left\{(v_{I1}(t) + jv_{Q1}(t))e^{j\omega_c t}\right\}$$

$$A(t) = \sqrt{v_{I1}^2(t) + v_{Q1}^2(t)}\ \text{, envelope of input signal}$$

$$P_A = 10\log\left(\frac{A^2(t)}{(2)(RIn)}\right) + 30\text{, power of envelope in dBm}$$

$$\varepsilon = SE\sin\left(2\pi\frac{(P_A - PMin)}{Ec}\right)$$

$$V_L = \sqrt{2}\left((RIn)10^{\frac{PMin - 30}{10}}\right)^{0.5}\ \text{, voltage level corresponding to PMin}$$

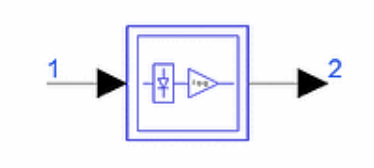
$$M(t) = \begin{cases} 20S\log\left(\frac{A(t)}{V_L}\right) + \varepsilon & \text{if } A(t) > V_L \\ 0 & \text{otherwise} \end{cases}$$

Then, the output signal $V_2(t)$ is given by the equation

$$V_2(t) = \frac{M(t)}{A(t)}V_1(t)$$

3. For information regarding timed nonlinear component signals, refer to *Timed Nonlinear Components* (timed).

LogVDet



Description: Logarithmic video detector
Library: Timed, Nonlinear
Class: TSDF_LogVDet
Derived: From baseOmniSysStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
Sensitivity	log sensitivity, in volts/dB	0	V	real	(0, ∞)
PMin	minimum input power, in dBm, for logarithmic amplification	-80		real	(-∞, ∞)
E	peak log error, in dB	0		real	[0, ∞)
Ec	log error cycle, in dB	0		real	(0, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
2	output	output signal	timed

Notes/Equations

1. LogVDet can be used to model a logarithmic video detector with some non-ideal behavior. The basic equation describing such a detector is:
   $$V_2(t) = K\log(V_1(t)/V_L) + \varepsilon$$
   where K determines the logarithmic slope, V_L determines the minimum input voltage required for logarithmic video detection, and ε is the deviation of the detector from the ideal.
2. The following equations describe the algorithm used for this model.
   Let
   $$V_1(t) = Re\left\{(v_{I1}(t) + jv_{Q1}(t))e^{j\omega_c t}\right\}$$

$$A(t) = \sqrt{v_{I1}^2(t) + v_{Q1}^2(t)}, \text{envelope of input signal}$$

$$P_A = 10\log\left(\frac{A^2(t)}{(2)(RIn)}\right) + 30, \text{power of envelope in dBm}$$

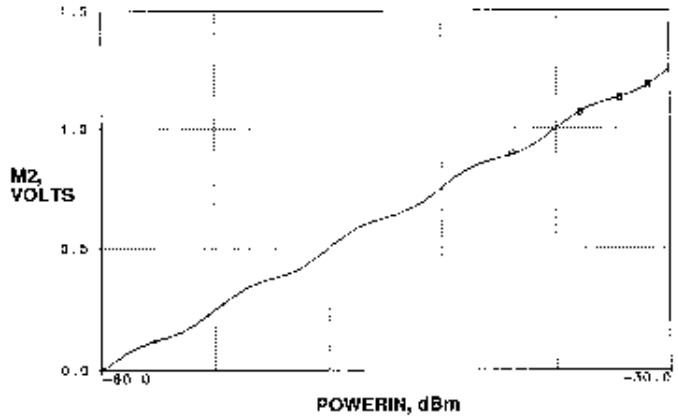
$$\varepsilon = SE\sin\left(2\pi\frac{(P_A - PMin)}{EC}\right)$$

$$V_L = \sqrt{2}\left((RIn)10^{\frac{PMin-30}{10}}\right)^{0.5}, \text{voltage level corresponding to PMin}$$

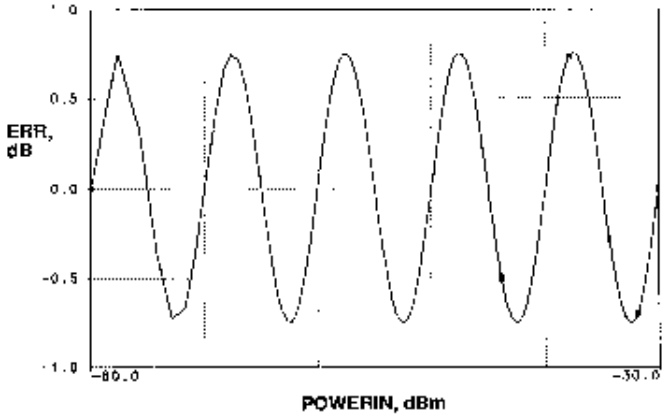
Then, the output signal $V_2(t)$ is given by the equation

$$V_2(t) = \begin{cases} 20S\log\left(\frac{A(t)}{V_L}\right) + \varepsilon & if A(t) > V_L \\ 0 & \text{otherwise} \end{cases}$$

Example: S= 0.025v/dB, PMin = −80 dBm, E = 0.75 dB, EC = 10 dB



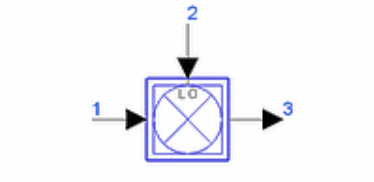
M2 vs. POWERIN



ERR vs. POWERIN

3. For information regarding timed nonlinear component signals, refer to *Timed Nonlinear Components* (timed).

MixerRF



Description: Complex single sideband mixer with RF and LO inputs
Library: Timed, Nonlinear
Class: TSDF_MixerRF
Derived From: baseNoiseFigureStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
NoiseFigure	input noise figure, in dB	0		real	[0, ∞)
Type	single sideband mixer type: RF plus LO, RF minus LO, LO minus RF	RF plus LO		enum	
RfRej	RF to IF rejection in dB	-200		real	(-∞, 0)
ImRej	Image rejection in dB	-200		real	(-∞, 0)
LoRej	LO rejection in dB	-200		real	(-∞, 0)
LComp	array of triple values for LO power (in dBm) vs gain change from small signal (in dB) and phase change from small signal (in degrees)	0 0 0		real array	Refer to Note 4

Pin Inputs

Pin	Name	Description	Signal Type
1	input1	input signal 1	timed
2	input2	input signal 2	timed

Pin Outputs

Pin	Name	Description	Signal Type
3	output	output signal	timed

Notes/Equations

1. MixerRF can be used to model a mixer with non-ideal characteristics. The non-ideal characteristics of the mixer include (LO power dependent) mixer gain, gain compression, and leakages from the RF signal, from the local oscillator, and from the image of the input signal.
   MixerRF must be used only when input signals are RF signals, for example, as an RF up- or down-converter. It is not intended to modulate a baseband signal, implying fc

> 0 for both input signals.

2. When leakage terms, such as RF to IF leakage, LO leakage, or image leakage are included, the bandwidth of the output signal can become very large, which will then require a very large sampling rate. This should be considered when including these terms.
3. The following equations describe the algorithm used for this model.
   The signals $V_k(t)$ at pins k = 1, 2 and 3 are represented by their inphase and quadrature components about the carrier frequency f_{ck} :

$$V_k(t) = Re\left\{ v_k(t) e^{j2\pi f_{ck} t} \right\}, \qquad k = 1,2,3$$

   The output signal $V_3(t)$ is determined as follows. The output carrier frequency is given by the equation

$$f_{c3} = \begin{cases} f_{c1} + f_{c2} & \text{if Type = RF plus LO} \\ |f_{c1} - f_{c2}| & \text{if Type = RF minus LO or LO minus RF} \end{cases}$$

   and the complex envelope of the output is given by the equation

$$v_3(t) = g_{mix} v_{sig} v_{lo} + v_{leakRF} + v_{leakLO} + v_{leakImage}$$

   where
   g_{mix} denotes mixer conversion gain (which can be made dependent on LO power by using the LComp data array)
   v_{sig} denotes input RF signal
   v_{lo} denotes local oscillator signal
   v_{leakRF} denotes leakage from RF rejection term
   v_{leakLO} denotes leakage from LO rejection term
   $v_{leakImage}$ denotes leakage from IF image rejection term
4. The above terms are determined as follows.
   Mixer conversion gain:

$$gmix = \begin{cases} 0.316 & \text{if LComp data is not specified} \\ g_i + jg_q & \text{if LComp data is specified} \end{cases}$$

   where the g_i and g_q values are interpolated from data specified in the LComp data array (interpolation is done for the small signal gain and angle corresponding to the instantaneous LO power). Note that the mixer conversion gain is determined with reference to the input LO power.
   The LComp data points are defined as an array. This array contains triple values of input power (dBm) small signal gain in dB and degrees.

```
LComp = "-30 -10 0 -20 -10.5 0 -15 -11 3 -12 -12 6 -11 -13 7 -10 -14.5 6"
```

   As an alternative from listing these data points, this dataset may be contained in a text file and referenced by name as follows:

```
LComp = "<mylcomp_data.re"
```

   where the *mylcomp_data.re* file must be located in the current workspace data subdirectory; if not in the subdirectory, then the name must include the full directory path as the prefix to the filename. The contents of this file simply triple values for each power point where the number separator can be a comma, space, tab, or new

line:
-30 -10 0
-20 -10.5 0
-15 -11 3
-12 -12 6
-11 -13 7
-10 -14.5 6

For details on setting and using arrays of data for parameter values, see *Array Parameters* (ptolemy) in the *ADS Ptolemy Simulation* (ptolemy) documentation.

Input RF signal:

$$v_{sig} = \begin{cases} v_{I1} - jv_{Q1} & \text{if Type=RF minus LO or LO minus RF and } (f_{c1} < f_{c2}) \\ v_{I1} + jv_{Q1} & \text{otherwise} \end{cases}$$

LO signal:

$$v_{lo} = \begin{cases} (v_{I2} - jv_{Q2})/|v_2| & \text{if Type=RF minus LO or LO minus RF and } (f_{c2} < f_{c1}) \\ (v_{I2} + jv_{Q2})/|v_2| & \text{otherwise} \end{cases}$$

RF rejection term:

$$v_{leakRF} = \begin{cases} 0 \text{ if } |\text{RFREJ}| < 200 \\ 10^{-\frac{|RFREJ|}{20}} \left\{ v_1 t e^{j2\pi f_{c1} - f_{c3} t} \right\} & \text{otherwise} \end{cases}$$

LO rejection term:

$$v_{leakLO} = \begin{cases} 0 \text{ if } |\text{LOREJ}| < 200 \\ 10^{-\frac{|LOREJ|}{20}} \left\{ v_2(t) e^{j2\pi(f_{c2} - f_{c3})t} \right\} & \text{otherwise} \end{cases}$$

IF image rejection term:

$$v_{leakImage} = \begin{cases} 0 \text{ if } |\text{IMREJ}| < 200 \\ 10^{\frac{|IMREJ|}{20}} \left\{ (v_{lo})(v_{I1} + jv_{Q1}) e^{j2\pi(f_{c1} + f_{c2} - f_{c3})t} \right\} & \text{if Type= RF- LO or LO - RF} \\ 10^{\frac{|IMREJ|}{20}} \left\{ (v_{lo}{}^*)(v_{I1} + jv_{Q1}) e^{j2\pi(|f_{c1} - f_{c2}| - f_{c3})t} \right\} & \text{if Type = RF + LO and } f_{c1} > f_{c2} \\ 10^{\frac{|IMREJ|}{20}} \left\{ (v_{lo})(v_{I1} + jv_{Q1}) e^{j2\pi(|f_{c1} - f_{c2}| - f_{c3})t} \right\} & \text{if Type = RF + LO and } (f_{c1} < f_{c2}) \end{cases}$$

where v_{lo*} - conjugate of v_{lo}

5. Noise may be contributed to the output signal from output resistor ROut and noise figure NoiseFigure. For no noise contribution from the resistor, set its temperature to (or below) absolute zero (−273.15). For no noise contribution from the noise figure, set its value to zero.
   Assuming matched source, load, and component input/output resistor conditions are all 50 ohms, the output noise power is given by
   N = N x G + k X T x B x G + k x T x B

$N_{out} = N_{in} G_p + k T_e B + k T_r B (G_p - 1)$

where

N_{out} = output noise power

N_{in} = input noise power

G_p = mixer power gain (=0.3162 if LComp data is not specified; depends on LComp data if LComp data is specified)

k = Boltzmann's constant = 1.38066e-23

T_e = equivalent noise temperature due to NoiseFigure = (F-1) x T_o

T_o = 290 Kelvin

$F = 10^{NoiseFigure/10}$

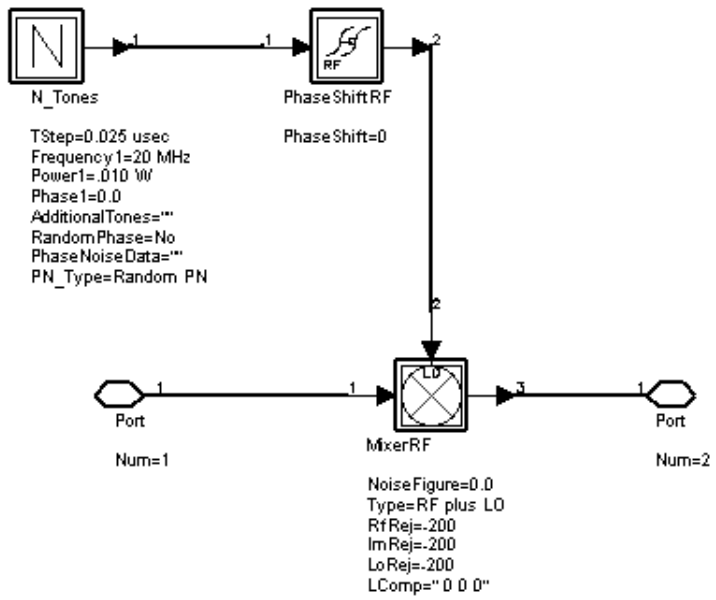
B = simulation bandwidth; 1/TStep (MixerRF accepts only RF signals at its input)

T_r = output resistor equivalent noise temperature = RTemp + 273.15

Appropriate scale factors apply for other source, load, and component input/output resistor conditions.

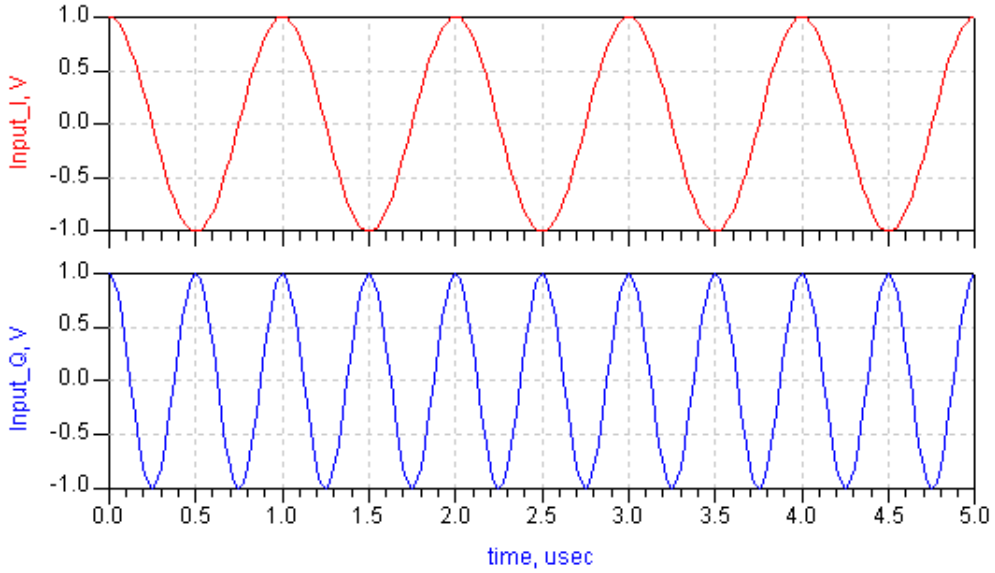
6. Schematic Using MixerRF shows a design in which MixerRF is used. Note that the complex phase shifter component PhaseShiftRF with phase parameter PhaseShift (in degrees) is used in the design to control the phase of the LO from the oscillator N_Tones component.

Schematic Using MixerRF



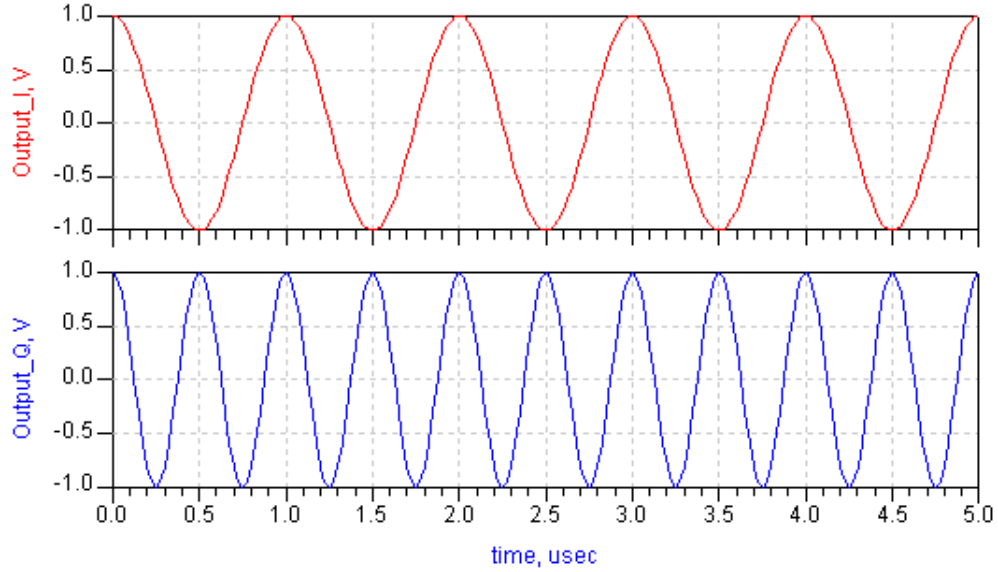
RF I- and Q-Channel Input at Pin 1 with FC1=40 MHz through RF I- and Q-Channel Output at Pin 3 show the input and output performance of MixerRF in the design schematic.

RF I- and Q-Channel Input at Pin 1 with FC1=40 MHz



RF I- and Q-Channel Output at Pin 3 shows MixerRF system component RF I- and Q-channel output at pin 3 using the following parameter values:
FC3 = 60 MHz
RF = 40 MHz
LO = 20 MHz
RfRej = −200
LoRej = −200
Type = RFplusLO
ImRej= −200
TStep = 0.025 µsec

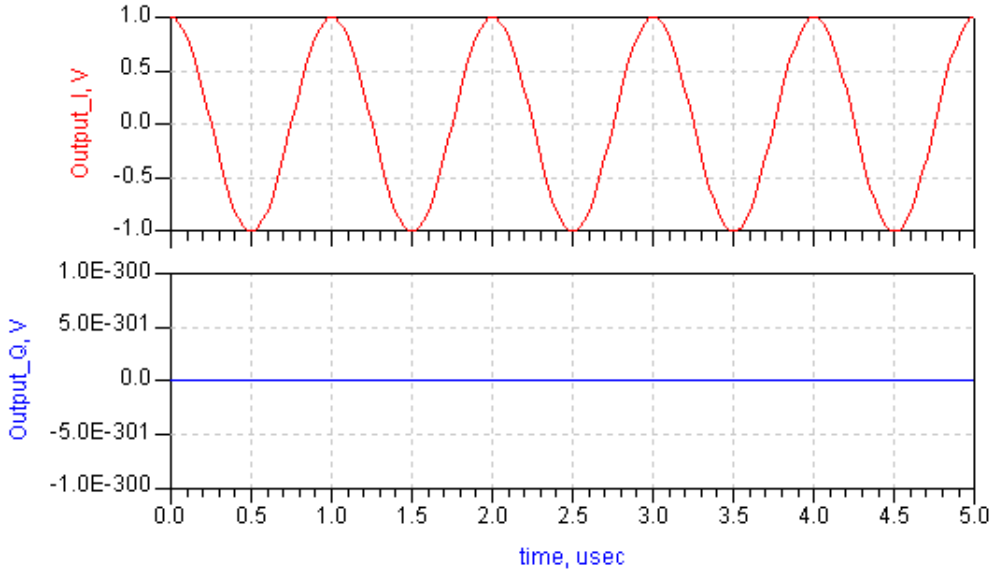
RF I- and Q-Channel Output at Pin 3



Baseband (RF minus LO) Output shows the MixerRF system component baseband (RF minus LO) output using the following parameter values:
FC3 = 0
RF = 40 MHz
LO = 40 MHz
RfRej = −200

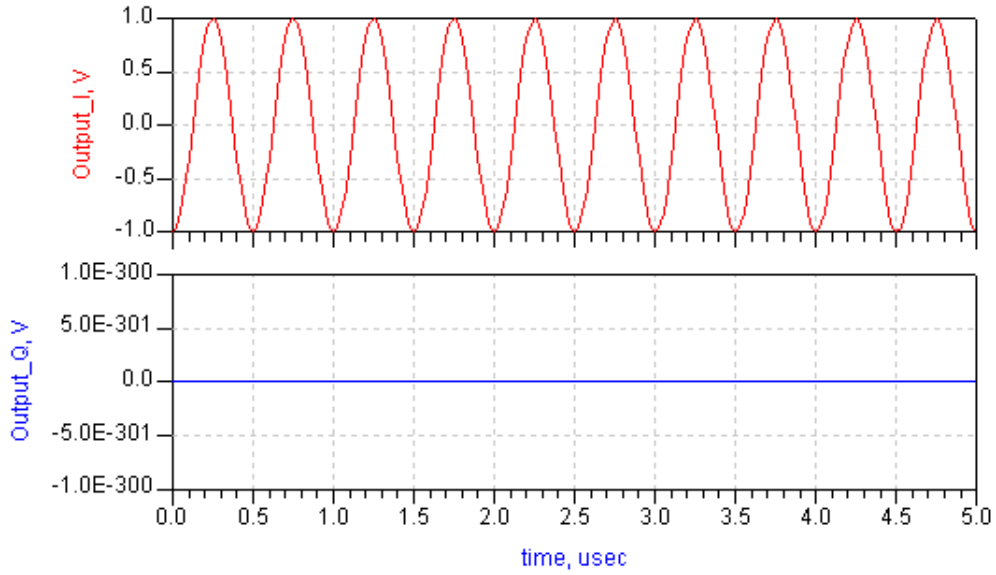
LoRej = −200
ImRej = −200
Type = RF minus LO
TStep = 0.025 µsec
PhaseShiftRF, PhaseShift = 0 degree
In Baseband (RF minus LO) Output, the Q output is 0V because the output signal is a baseband signal, that is, FC3=0.

Baseband (RF minus LO) Output



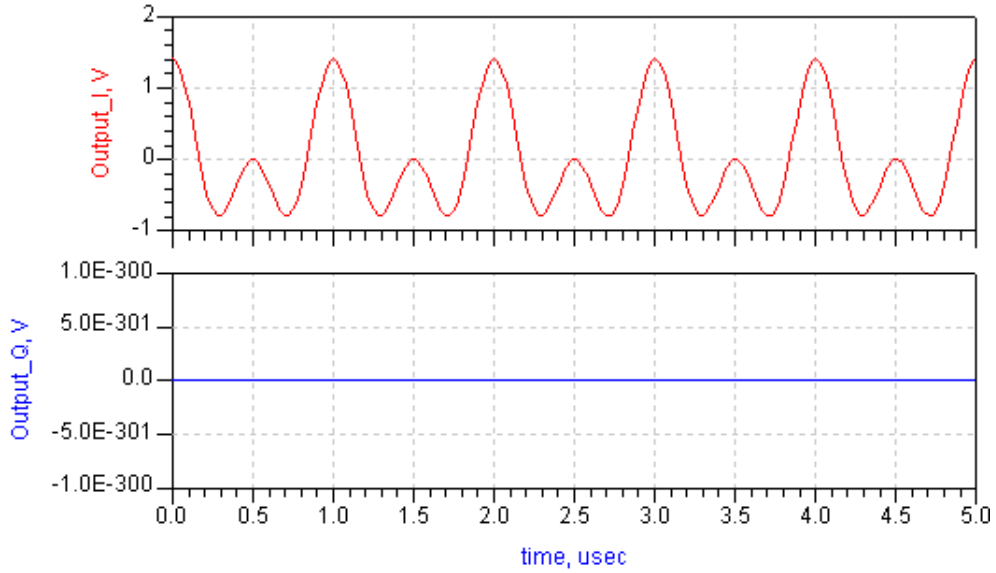
Baseband (RF minus LO) Output shows the MixerRF system component baseband (RF minus LO) output using the following parameter values:
FC3 = 0
RF = 40 MHz
LO = 40 MHz
RfRej = −200
LoRej = −200
ImRej = −200
Type = RF minus LO
TStep = 0.025 µsec
PhaseShiftRF, PhaseShift = −90 degree
In Baseband (RF minus LO) Output, the Q output is 0V because the output signal is a baseband signal, that is, FC3=0.

Baseband (RF minus LO) Output



Baseband (RF minus LO) Output shows the MixerRF system component baseband (RF minus LO) output using the following parameter values:
FC3 = 0
RF = 40 MHz
LO = 40 MHz
RfRej = −200
LoRej = −200
ImRej = −200
Type = RF minus LO
TStep = 0.025 µsec
PhaseShiftRF, PhaseShift = 45 degrees
In Baseband (RF minus LO) Output, the Q output is 0V because the output signal is a baseband signal, that is, FC3=0.

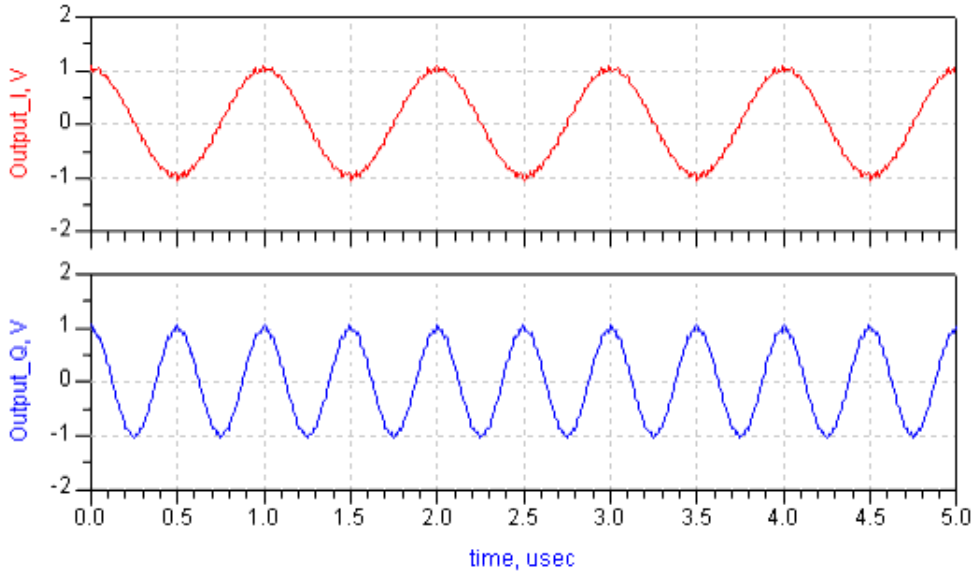
Baseband (RF minus LO) Output



RF I- and Q-Channel Output at Pin 3 shows the MixerRF system component RF I- and Q-channel output at pin 3 using the following parameter values:

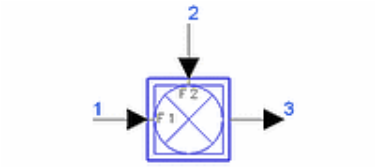
FC3 = 20 MHz
RF = 40 MHz
LO = 20 MHz
RfRej = −30
LoRej = −30
ImRej = −30
Type = RF minus LO
TStep = 0.0025 µsec
PhaseShiftRF, PhaseShift = 0 degree

RF I- and Q-Channel Output at Pin 3



7. For information regarding timed nonlinear component signals, refer to *Timed Nonlinear Components* (timed).

MultiplierRF



Description: Two-input RF signal multiplier
Library: Timed, Nonlinear
Class: TSDF_MultiplierRF
Derived From: baseStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
Type	selected multiplication product: F1+F2 Upper sideband, F1-F2 Lower sideband, Both sidebands	F1+F2 Upper sideband		enum	

Pin Inputs

Pin	Name	Description	Signal Type
1	input1	input signal 1	timed
2	input2	input signal 2	timed

Pin Outputs

Pin	Name	Description	Signal Type
3	output	output signal	timed

Notes/Equations

1. MultiplierRF can be used to model an up-converter, down-converter, or double sideband modulator. The following points should be noted.
   - For Type=Both: either input can be an RF complex envelope or baseband representation.
   - For Type=F1+F2 or F1−F2: both inputs must be an RF complex envelope representation.
   - When both inputs have a baseband representation, the output is equal to the product of the inputs, that is, Type is effectively set equal to all.
   - When both inputs have a complex envelope representation at different carrier frequencies and Type is set to F1−F2, and f1 is not equal to f2, the output will have a complex envelope representation at the difference of the two input carrier frequencies. This complex envelope representation is invalid if the output signal is a lowpass signal, that is, if it has significant energy at 0 Hz. It is then the responsibility of the user to change the representation of the signal to a baseband representation (which can be done by placing an FcChange component at the output with parameter FcOut equal to 0). This comment also applies if

Type is set equal to all, and the lower sideband of the product is a lowpass signal.

2. The following equations describe the algorithm for this model.
   The signals $V_k(t)$ at pins k = 1, 2 and 3 are represented by their inphase and quadrature components about carrier frequency f_{ck} :

$$V_k(t) = Re\left\{ v_k(t) e^{j2\pi f_{ck} t} \right\}, \quad v_k(t) = v_{Ik}(t) + j v_{Qk}(t), \quad k = 1, 2, 3$$

   Output signal V_3 (t) is determined as follows. Output carrier frequency is given by the equation:

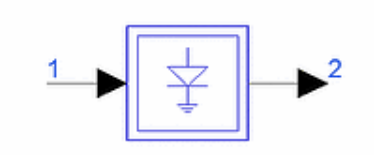
$$f_{c3} = \begin{cases} f_{c1} + f_{c2} & \text{if Type} = \text{F1} + \text{F2} \\ |f_{c1} - f_{c2}| & \text{if Type} = \text{F1 - F2} \\ \max(f_{c1}, f_{c2}) & \text{if Type} = \text{Both} \end{cases}$$

$$v_3(t) = \begin{cases} v_1(t) v_2(t) & \text{if } f_{c1} = f_{c2} = 0 \\ \frac{1}{2} v_1(t) v_2(t) & \text{if Type} = \text{F1} + \text{F2} \\ \frac{1}{2} v_1(t) v_2^*(t) & \text{if Type} = \text{F1 - F2 and } f_{c1} \geq f_{c2} \\ \frac{1}{2} v_1^*(t) v_2(t) & \text{if Type} = \text{F1 - F2 and } f_{c1} < f_{c2} \\ \frac{1}{2} v_1(t) v_2(t) + \frac{1}{2} v_1(t) v_2^*(t) & \text{if Type} = \text{Both} \end{cases}$$

   where * denotes the complex conjugate of the signal.

3. For information regarding timed nonlinear component signals, refer to *Timed Nonlinear Components* (timed).

PeakDetector



Description: Peak RF detector
Library: Timed, Nonlinear
Class: TSDF_PeakDetector
Derived From: baseOmniSysStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
DroopRate	output voltage decay rate in fractional volts per second	0.1		real	[0, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
2	output	output signal	timed

Notes/Equations

1. PeakDetector can be used to model a peak detector. Its output is equal to the previous maximum input value. It has a droop parameter to model non-ideal behavior where the output value decays at a constant rate determined by the parameter DroopRate.
2. The following equations describe the algorithm used for this model.
   Let
   $V'_m(t) = V_m(t)\{1 - [t - T_m(t)]\,|\text{DroopRate}|\}$
   where
   T_m is the time instant at which the previous maximum value occurred

$$V_m(t) = \begin{cases} -\infty & \text{at } t = 0 \\ V_1(t) & \text{if } V_1(t) > V'_m(t - T_S) \\ V_m(t - T_S) & \text{otherwise} \end{cases}$$

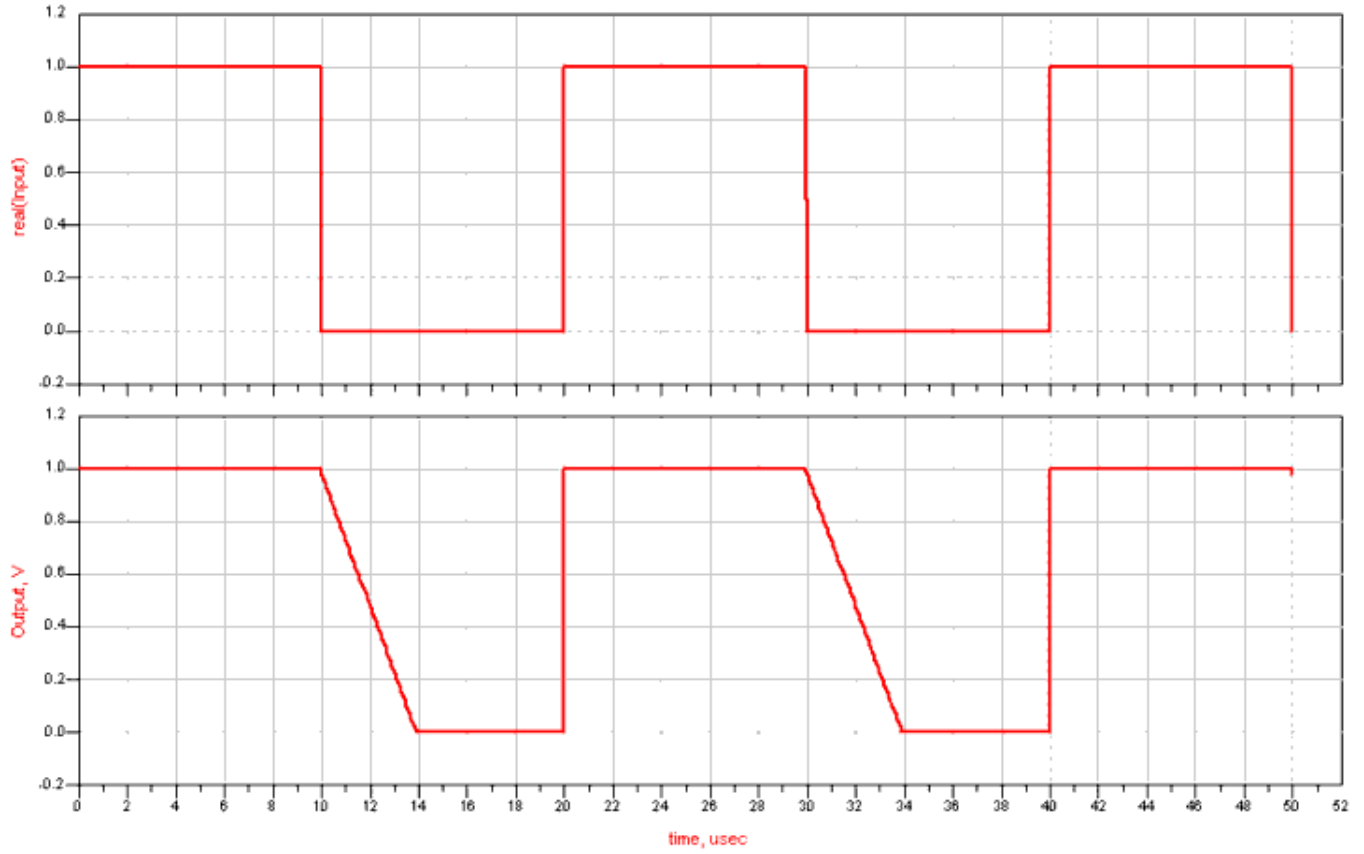
$$T_m(t) = \begin{cases} 0 & \text{at t = 0} \\ t & \text{if } V_1(t) > V'_m(t - T_S) \\ T_m(t - T_S) & \text{otherwise} \end{cases}$$

T_S is the sampling period of the input signal

Then

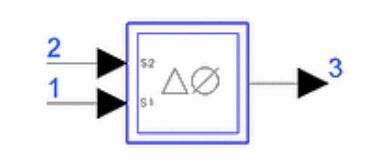
$$V_2(t) = \begin{cases} V_1(t) & if\ V_1(t) > V'_m(t) \\ V'_m(t) & \text{otherwise} \end{cases}$$

3. The input and output signals for a PeakDetector component with DroopRate=1V/(4µsec) are shown.



4. For information regarding timed nonlinear component signals, refer to *Timed Nonlinear Components* (timed).

PhaseCompRF



Description: RF phase comparator
Library: Timed, Nonlinear
Class: TSDF_PhaseCompRF
Derived From: baseOmniSysStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
PhaseCharacteristicType	type of analog phase comparator: Linear, Sinusoidal, Tringular	Linear		enum	
GainConstant	small signal gain constant, in volts per degree	1		real	(-∞, 0) or (0, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	s1	input signal 1	timed
2	s2	input signal 2	timed

Pin Outputs

Pin	Name	Description	Signal Type
3	output	output signal	timed

Notes/Equations

1. PhaseCompRF models a phase comparator with linear, sine and triangular characteristics.

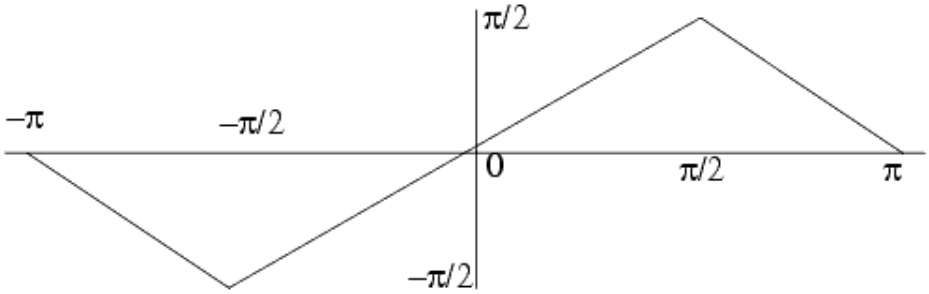
2. Let $\theta_1(t)$ denote the phase angle associated with $V_1(t)$ the input signal at pin 1; let $\theta_2(t)$ denote the phase angle associated with $V_2(t)$ the input signal at pin 2.

   Let K = GainConstant. Then, the output signal at pin 3 is:

$$V_3(t) = \begin{cases} K\frac{180}{\pi}(\theta_1 - \theta_2) & for\ \text{Type} = \text{Linear} \\ K\frac{180}{\pi}\sin(\theta_1 - \theta_2) & for\ \text{Type} = \text{Sine} \\ K\frac{180}{\pi}triangular(\theta_1 - \theta_2) & for\ \text{Type} = \text{Triangular} \end{cases}$$

where the triangular() function is shown below.

Phase Comparator Characteristics



3. The following equations describe the algorithm used for this model. (A fundamental assumption is that the input signals are represented by their complex envelopes.) Let the input signal at pin 1 be denoted as

$$V_1(t) = Re\left\{(v_{I1}(t) + jv_{Q1}(t))e^{j2\pi f_{c1}t}\right\}$$

and the input signal at pin 2 be denoted as

$$V_2(t) = Re\left\{(v_{I2}(t) + jv_{Q2}(t))e^{j2\pi f_{c2}t}\right\}$$

Then the phase angle associated with the input signal at pin 1 is

$$\theta_1 = \tan^{-1}\left(\frac{v_{Q1}(t)}{v_{I1}(t)}\right)$$

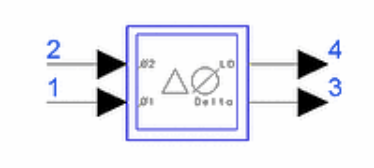
And, the phase angle associated with the input signal at pin 2 is

$$\theta_2 = \tan^{-1}\left(\frac{Im\left((v_{I2}(t) + jv_{Q2}(t))e^{j2\pi(f_{c2}-f_{c1})t}\right)}{Re\left((v_{I2}(t) + jv_{Q2}(t))e^{j2\pi(f_{c2}-f_{c1})t}\right)}\right)$$

The output signal at pin 3 is then determined using these phase angles and the phase characteristics as described in note 2.

4. This component is useful in RF phase lock loops. For control loop details, refer to *Second-Order Control Loop Filters* (timed) in the *Timed Filters* (timed) documentation.

PhaseFreqComp



Description: Phase/frequency comparator, digital 3-state type
Library: Timed, Nonlinear
Class: TSDF_PhaseFreqComp
Derived From: baseOmniSysStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
LoLevel	low voltage level	-1	V	real	(-∞, ∞)
HiLevel	high voltage level	1	V	real	(LoLevel, ∞)
UnLockR	internal phase detector output resistance level when not phase locked	0.0001	Ohm	real	[0, ∞)†
LockR	internal phase detector output impedance level when phase locked	1000000	Ohm	real	(0, ∞)
FilterR1	internal lowpass filter series resistor	2.17	Ohm	real	[0, ∞)†
FilterR2	internal lowpass filter shunt resistor	1.83	Ohm	real	[0, ∞)†
FilterC	internal lowpass filter shunt capacitor	0.000000000001	F	real	(0, ∞)

† UnlockR + FilterR1 + FilterR2 > 0

Pin Inputs

Pin	Name	Description	Signal Type
1	phase1	input signal 1	timed
2	phase2	input signal 2	timed

Pin Outputs

Pin	Name	Description	Signal Type
3	delta	phase difference level	timed
4	LD	lock detector signal	timed

Notes/Equations

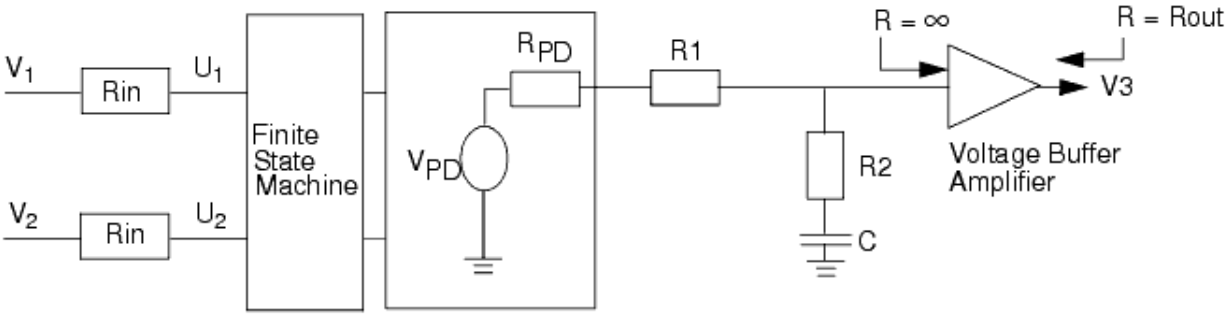
1. PhaseFreqComp is a 3-state output phase/frequency comparator similar to Phase Comparator 2 on the Motorola MC14046 IC.
   PhaseFreqComp Model shows the model that is implemented.
   The value of V_{PD} at any instant depends on the value of the present input voltages, the past input voltages, and the previous value of V_{PD}. The state transition diagram shown in PhaseFreqComp State Transition Diagram of the Finite State Machine specifies this relationship.
   Initially, the component is in the 00 state of the lock region.
   Let Vl = LoLevel, Vh = HiLevel, Rl = UnLockR, Rh = LockR, R1 = FilterR1, R2 = FilterR2, C = FilterC

   Important
   Voltages V1 and V2 are considered to be a logic 0 or logic 1 state depending on whether the voltages are greater than or less than V_{TH} where V_{TH} =(Vl_Vh)/2.

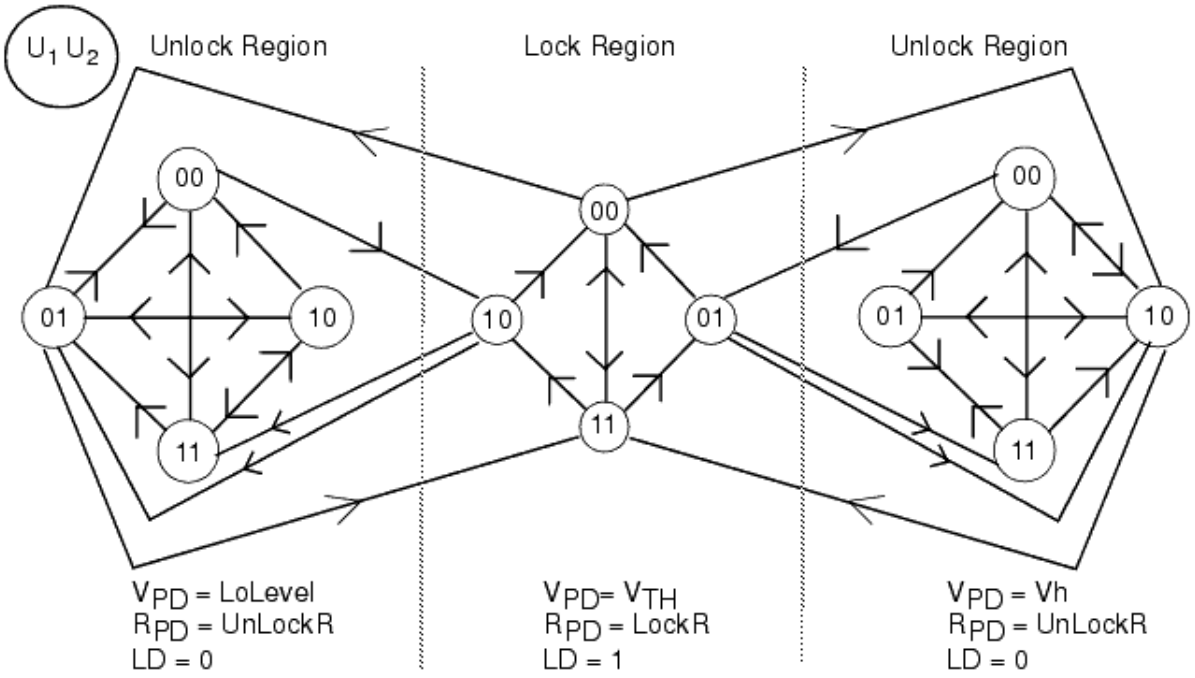
   The value of resistance R_{PD} is equal to Rh (that is, the output is in the high impedance state) when the component is in the lock region; when the component is in the unlock region, the value of R_{PD} is equal to Rl.

   The network formed by R1, R2, and C provides a transfer function with a zero at 1/(R2 × C) and a pole at 1/((R1 + R2) × C) and can be used as a loop filter in a phase-lock-loop (PLL) circuit. If an external loop filter is desired, the values of R1, R2 and C can be adjusted so that the pole and zero fall at a frequency far outside the passband of the external filter.
   V_4 (t) is used as a lock detector (LD) when this component is used in a PLL application. V4 is equal to 1.0V units when the component is in the lock region and equal to 0V otherwise.

PhaseFreqComp Model

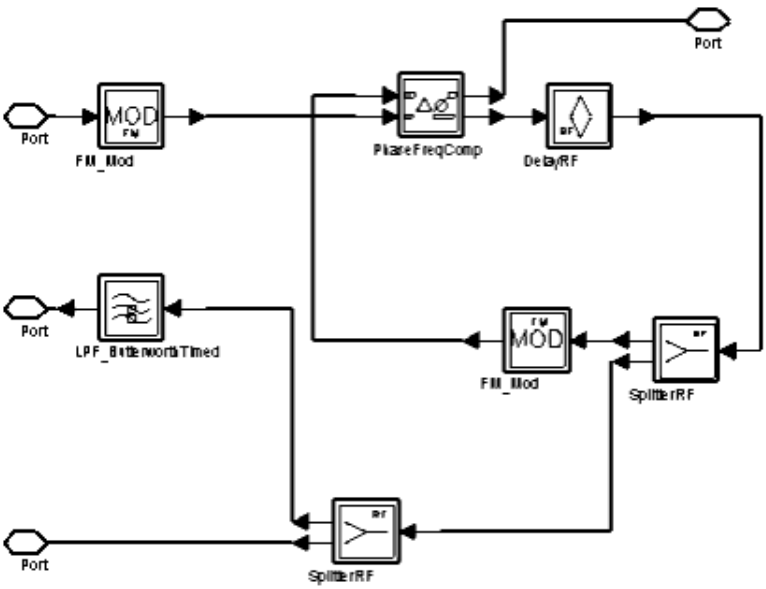


PhaseFreqComp State Transition Diagram of the Finite State Machine

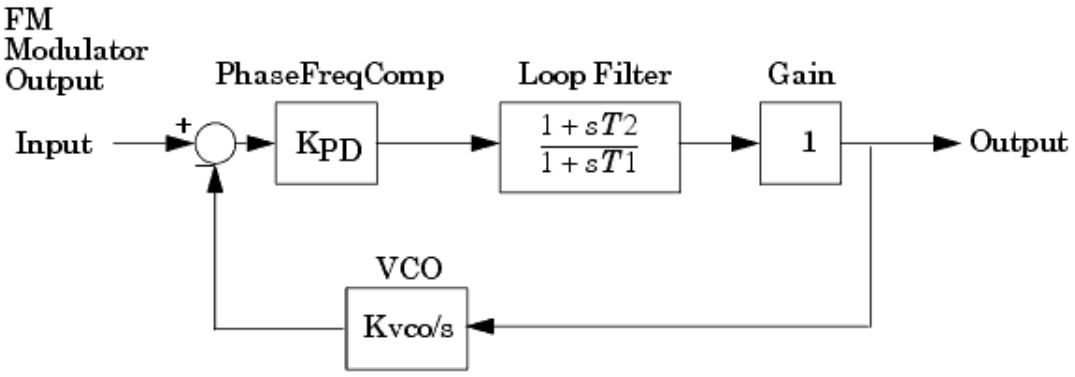


1. Consider the PLL application example in PLL with PhaseFreqComp using PhaseFreqComp to implement the phase detector and loop filter portion of the PLL. In PLL with PhaseFreqComp, the PLL is used to provide a phase coherent clock signal from its input. The input signal is an FM signal (±1V) modulated by a step function of maximum magnitude of 0.5V. PhaseFreqComp acts as a phase frequency detector that compares the input signal with the PLL VCO signal and generates an error signal for the control loop.
   A linear analysis of the PLL is performed by considering the linear model representation of the PLL as shown in PLL Linear Model.

PLL with PhaseFreqComp



PLL Linear Model



The phase detector gain K_{PD} is equal to (Vh − Vl)/(4PI) (refer to Motorola MC14046B specifications). The VCO gain Kvco is equal to Sensitivity (2π)/s, where Sensitivity is Hz/volt. The loop filter (built into PhaseFreqComp) is represented as the transfer function:

$$H(s) = \frac{(R_2Cs+1)}{((R_1+R_2)Cs+1)} = \frac{(1+sT_2)}{(1+sT_1)}$$

T1=4 µsec and T2=1.83 µsec.
From a comparison of the denominator and the standard form

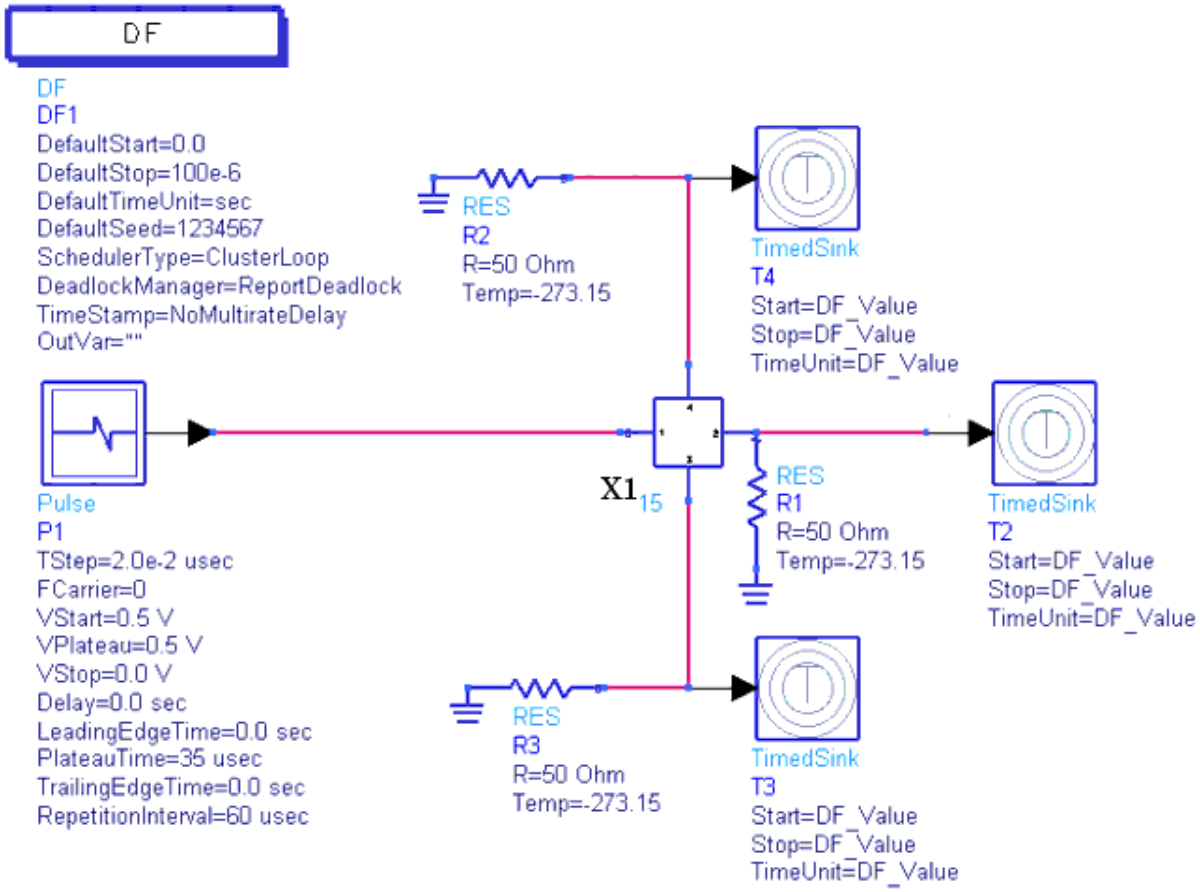
$$s^2 + 2\omega_n\xi s + \omega_n^2$$

we see that the loop natural frequency is 0.5 Mrad/s and the damping factor is 0.707. For control loop details, refer to *Second-Order Control Loop Filters* (timed) in the *Timed Filters* (timed) documentation.

Remarks

Design of a PLL with a phase frequency comparator such as PhaseFreqComp is usually done using a linear model for analysis. Because PhaseFreqComp is nonlinear, a linear analysis is an approximation and differences will be evident between the actual and the linear model performance.
PLL Schematic (X1= PLL with PhaseFreqComp) shows the associated schematic for the PLL example. A periodic square waveform signal with low value of 0V and high value of 0.5V is input into the PLL design shown in PLL Linear Model.

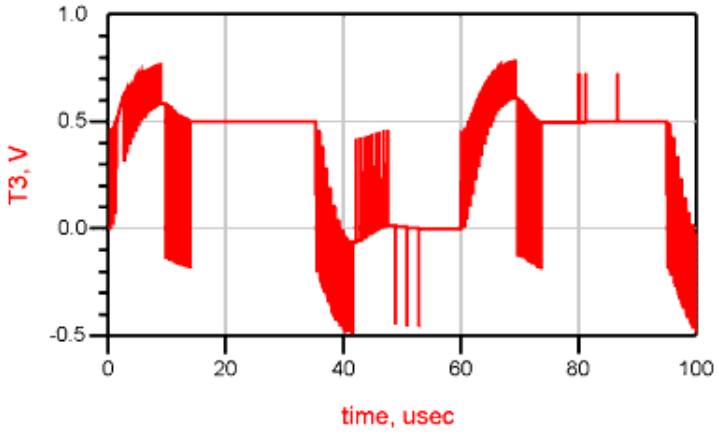
PLL Schematic (X1= PLL with PhaseFreqComp)



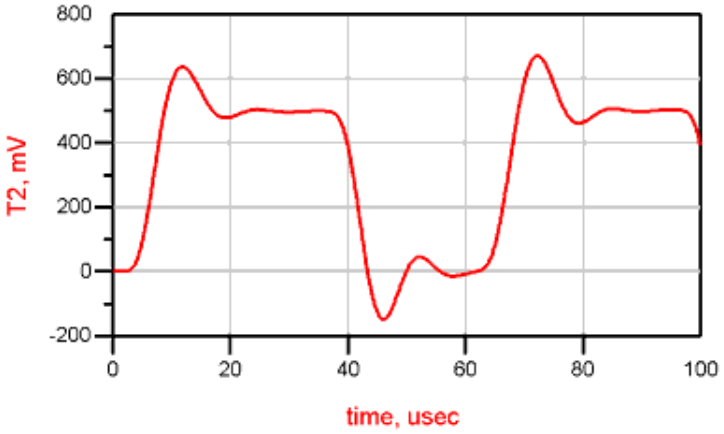
Step Response of PLL using PhaseFreqComp and *5-17* show the response of the PLL design example to the square wave input (STEP_INPUT). Step Response of PLL using PhaseFreqComp shows the PLL output without any smoothing (lowpass filtering) of the output signal (PLL_OUTPUT). Note the presence of the high-frequency switching due to the three values taken by the PhaseFreqComp Vpd voltage (see the model for PhaseFreqComp shown in PhaseFreqComp Model). Step Response of PLL (with LPF) using PhaseFreqComp shows the PLL output with lowpass filtering (PLL_OUTPUT_WITH_LPF). LD Output of PhaseFreqComp in PLL Example shows the LD (Lock Detector) output of the PhaseFreqComp component.

From Step Response of PLL using PhaseFreqComp, *5-17*, and *5-18* it can be seen that the LD output (Lock_Detector_Output) indicates when the PLL is in lock by going high (1.0V). When the PLL is still in the process of acquiring lock (note the transition regions in LD Output of PhaseFreqComp in PLL Example), the LD output will switch frequently between 0V and 1V as the PhaseFreqComp component attempts to correct for phase error. Therefore, lock detection can be accomplished by comparing the average LD output with a threshold value.

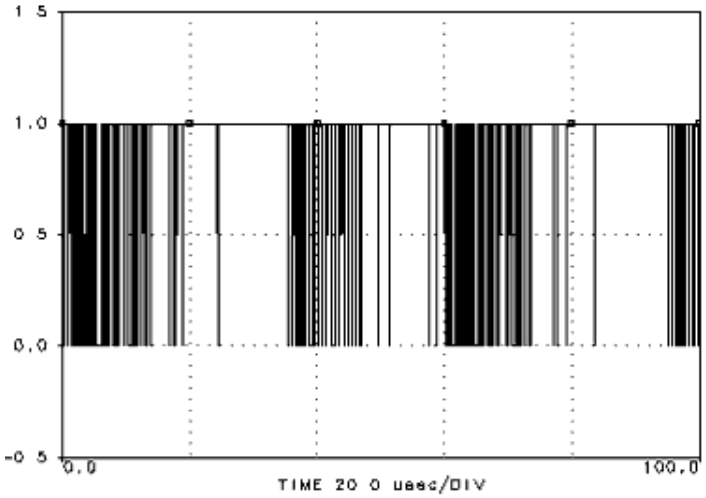
Step Response of PLL using PhaseFreqComp



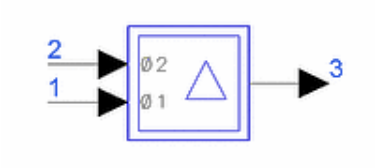
Step Response of PLL (with LPF) using PhaseFreqComp



LD Output of PhaseFreqComp in PLL Example



PhaseFreqCompXOR



Description: Phase/frequency comparator, exclusive OR type
Library: Timed, Nonlinear
Class: TSDF_PhaseFreqCompXOR
Derived From: baseOmniSysStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
LoLevel	low voltage level	0	V	real	(-∞, ∞)
HiLevel	high voltage level	1	V	real	(LoLevel, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	phase1	input signal 1	timed
2	phase2	input signal 2	timed

Pin Outputs

Pin	Name	Description	Signal Type
3	output	output signal	timed

Notes/Equations

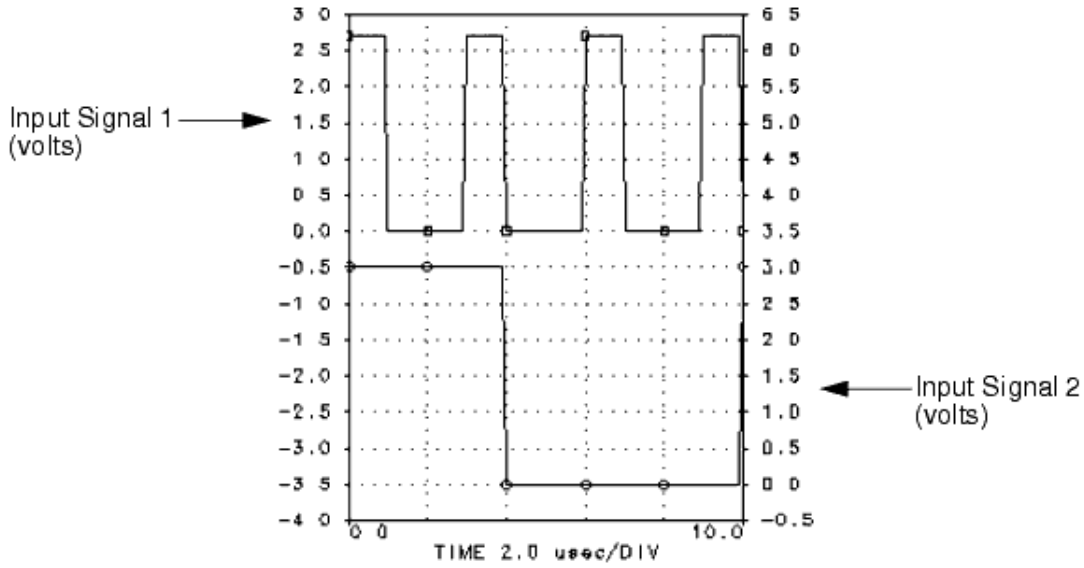
1. PhaseFreqCompXOR acts like an exclusive OR gate with levels that can be set by the user. This is the same as Phase Comparator 1 on the MC14046 IC.
2. Let Vl = LoLevel, Vh = HiLevel, V_{TH} = (Vl + Vh)/2

$$V_3(t) = \begin{cases} Vh & \text{if } V_1(t) \le V_{TH} \text{ and } V_2(t) > V_{TH} \\ Vh & \text{if } V_1(t) > V_{TH} \text{ and } V_2(t) \le V_{TH} \\ Vl & \text{if } V_1(t) \le V_{TH} \text{ and } V_2(t) \le V_{TH} \\ Vl & \text{if } V_1(t) > V_{TH} \text{ and } V_2(t) > V_{TH} \end{cases}$$

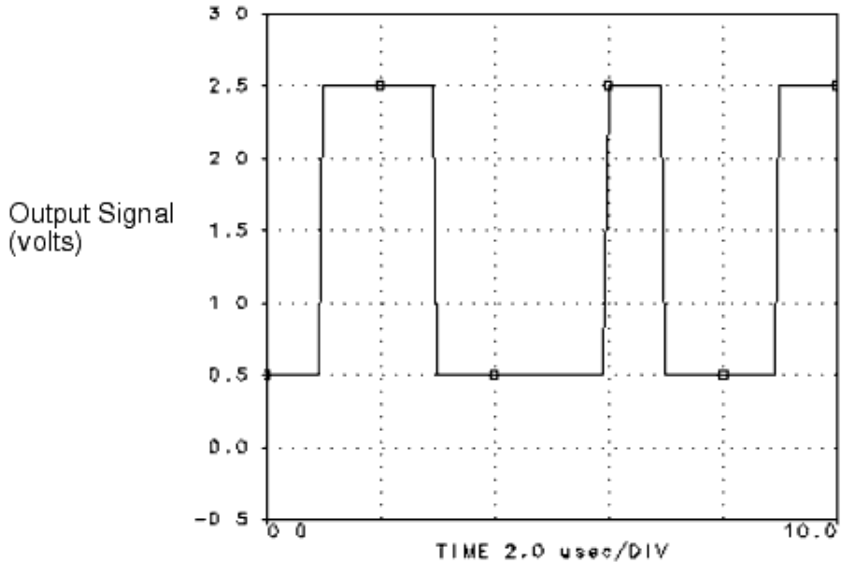
3. The signal voltages to the PhaseFreqCompXOR component are shown in PhaseFreqCompXOR Input Signal Voltages; parameters are Vl=0.5V and Vh=2.5V. The output signal of PhaseFreqCompXOR is shown in PhaseFreqCompXOR Output Signal (given the input signals in PhaseFreqCompXOR Input Signal Voltages).
4. This component is useful in RF phase lock loops. For control loop details, refer to *Second-Order Control Loop Filters* (timed) in the *Timed Filters* (timed)

documentation.

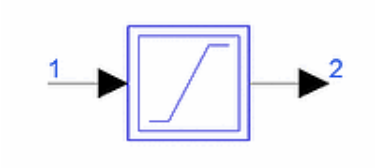
PhaseFreqCompXOR Input Signal Voltages



PhaseFreqCompXOR Output Signal



RateLimiter



Description: Signal rate of change limiter
Library: Timed, Nonlinear
Class: TSDF_RateLimiter
Derived From: baseOmniSysStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
RMax	maximum slew rate in volts per second	1		real	(0, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
2	output	output signal	timed

Notes/Equations

1. Many components such as logic gates and flip-flops are modeled as ideal components whose outputs have infinite rise and fall times. RateLimiter can be used in such cases to give the signals finite rise and fall times. RMax can be used to set the maximum rate at which the signal can change.
2. The output signal is always a baseband signal.
3. The following algorithm is used to determine the output signal $V_2(t)$.

   Let

   $$m = \frac{V_1(t) - V_1(t - T_S)}{T_S}$$

   where T_S is the sampling period of the input signal.

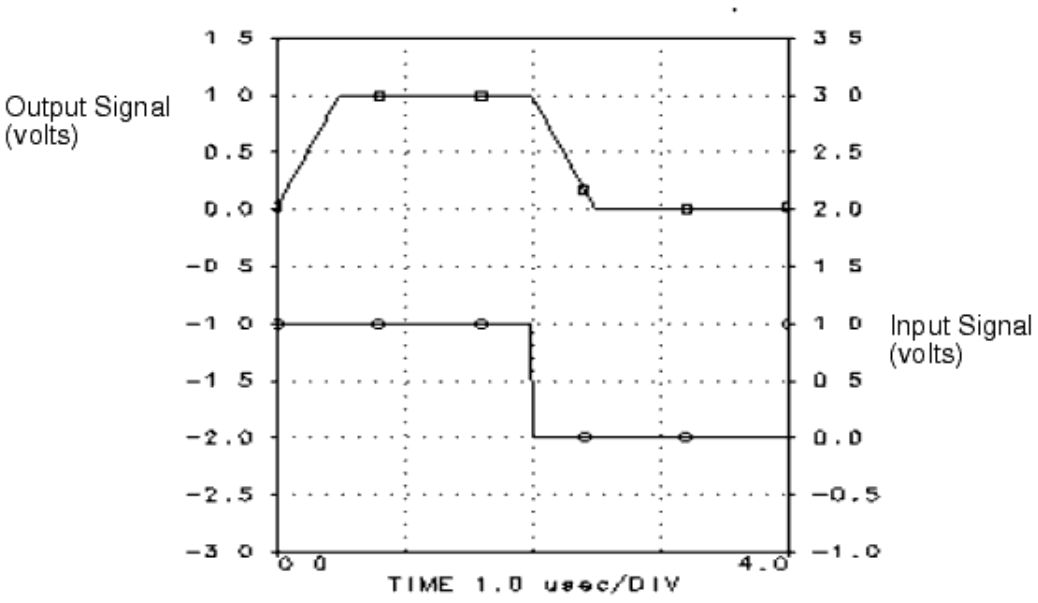
   Then

   $$V_2(t) = \begin{cases} V_1(t) & \text{if} |m| \le Rmax \\ V_2(t - TS) + \text{sgn}(m) T_S \, Rmax & \text{if} |m| > Rmax \end{cases}$$

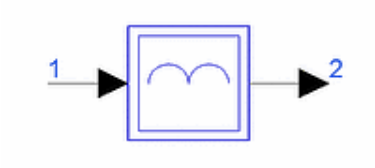
4. The input and output signal voltages, with parameter Rmax=2V/µsec, are shown in

RateLimiter Signal Plot. The TStep is 0.01 µsec.

RateLimiter Signal Plot



Rectifier



Description: Signal rectifier
Library: Timed, Nonlinear
Class: TSDF_Rectifier
Derived From: baseOmniSysStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
Type	type of signal rectifier: Full wave rectifier, Half wave rectifier	Full wave rectifier		enum	

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
2	output	output signal	timed

Notes/Equations

1. When Type = Full wave rectifier, the output is represented as:

$$V_2(t) = Re\left\{(v_{I2}(t) + jv_{Q2}(t))e^{j\omega_c t}\right\}$$

where
$v_{I2}(t) = |V_1(t)|$
$v_{Q2}(t) = 0$
$\omega c = 0$
When Type = Half wave rectifier, the output is represented as:

$$V_2(t) = Re\left\{(v_{I2}(t) + jv_{Q2}(t))e^{j\omega_c t}\right\}$$

where
$v_{I1}(t) = V_1(t)$ if $V_1(t) > 0$

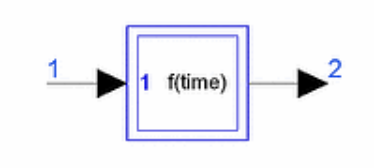
$v_{I2}(t) = 0$ if $V_1(t) \leq 0$

$v_{Q2}(t) = 0$

fc2 = 0

2. For information regarding timed nonlinear component signals, refer to *Timed Nonlinear Components* (timed).

TimedSDC1



Description: Timed Symbolic Defined Component
Library: Timed, Nonlinear
Class: TSDFTimedSDC1

Parameters

Name	Description	Default	Unit	Type
FCarrier	Carrier frequency expression, function of inputs' carrier frequencies	0.0	Hz	real
Expression	Expression, function of inputs and "time"	0.0+j*0.0		complex
RIn	input resistance	DefaultRIn	Ohm	real
ROut	output resistance	DefaultROut	Ohm	real
RTemp	physical temperature in degrees C	DefaultRTemp	Celsius	real

Pin Inputs

Pin	Name	Description	Signal Type
1	input1	input signal 1	timed

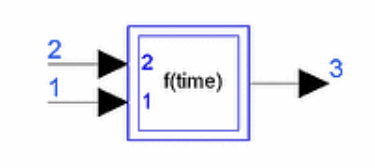
Pin Outputs

Pin	Name	Description	Signal Type
2	output	output signal	timed

Notes/Equations

1. This component generates a timed data output that is evaluated using an expression. Expression can be any valid expression, following the syntax used for writing expressions on a VAR block.
   VAR exception: The VAR expression must not be based on a Data Access component. To use the Data Access component use one of the numeric SDC components (SDC1, ..., SDC4).
2. Input data is specified by predefined variables _v1, _v2, _fc1, _fc2, etc. where v specifies signal voltages and fc specifies carrier frequencies; 1 and 2 are the port numbers.
3. The Expression can be also be dependent on time and _fc, where time is the simulation time that is incremented by Tstep for each firing of this component decided by the schedule and _fc is the resultant of FCarrier expression.
4. The FCarrier expression can be dependent on _fc1, _fc2, etc.
5. When FCarrier is 0, the imaginary part of the Expression will be ignored.

TimedSDC2



Description: Timed Symbolic Defined Component
Library: Timed, Nonlinear
Class: TSDFTimedSDC2
Derived From: TimedSDC1

Parameters

Name	Description	Default	Unit	Type
FCarrier	Carrier frequency expression, function of inputs' carrier frequencies	0.0	Hz	real
Expression	Expression, function of inputs and "time"	0.0+j*0.0		complex
RIn	input resistance	DefaultRIn	Ohm	real
ROut	output resistance	DefaultROut	Ohm	real
RTemp	physical temperature in degrees C	DefaultRTemp	Celsius	real

Pin Inputs

Pin	Name	Description	Signal Type
1	input1	input signal 1	timed
2	input2	input signal 2	timed

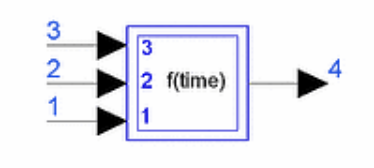
Pin Outputs

Pin	Name	Description	Signal Type
3	output	output signal	timed

Notes/Equations

1. This component generates a timed data output that is evaluated using an expression. Expression can be any valid expression, following the syntax used for writing expressions on a VAR block.
   VAR exception: The VAR expression must not be based on a Data Access component. To use the Data Access component use one of the numeric SDC components (SDC1, ..., SDC4).
2. Input data is specified by predefined variables _v1, _v2, _fc1, _fc2, etc. where v specifies signal voltages and fc specifies carrier frequencies; 1 and 2 are the port numbers.
3. The Expression can be also be dependent on time and _fc, where time is the simulation time that is incremented by Tstep for each firing of this component decided by the schedule and _fc is the resultant of FCarrier expression.
4. The FCarrier expression can be dependent on _fc1, _fc2, etc.
5. When FCarrier is 0, the imaginary part of the Expression will be ignored.

TimedSDC3



Description: Timed Symbolic Defined Component
Library: Timed, Nonlinear
Class: TSDFTimedSDC3
Derived From: TimedSDC2

Parameters

Name	Description	Default	Unit	Type
FCarrier	Carrier frequency expression, function of inputs' carrier frequencies	0.0	Hz	real
Expression	Expression, function of inputs and "time"	0.0+j*0.0		complex
RIn	input resistance	DefaultRIn	Ohm	real
ROut	output resistance	DefaultROut	Ohm	real
RTemp	physical temperature in degrees C	DefaultRTemp	Celsius	real

Pin Inputs

Pin	Name	Description	Signal Type
1	input1	input signal 1	timed
2	input2	input signal 2	timed
3	input3	input signal 3	timed

Pin Outputs

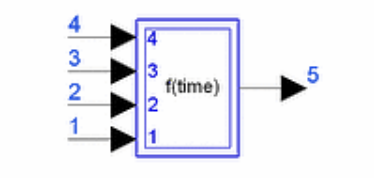
Pin	Name	Description	Signal Type
4	output	output signal	timed

Notes/Equations

1. This component generates a timed data output that is evaluated using an expression. Expression can be any valid expression, following the syntax used for writing expressions on a VAR block.
   VAR exception: The VAR expression must not be based on a Data Access component. To use the Data Access component use one of the numeric SDC components (SDC1, ..., SDC4).
2. Input data is specified by predefined variables _v1, _v2, _fc1, _fc2, etc. where v specifies signal voltages and fc specifies carrier frequencies; 1 and 2 are the port numbers.
3. The Expression can be also be dependent on time and _fc, where time is the simulation time that is incremented by Tstep for each firing of this component decided by the schedule and _fc is the resultant of FCarrier expression.

4. The FCarrier expression can be dependent on _fc1, _fc2, etc.
5. When FCarrier is 0, the imaginary part of the Expression will be ignored.

TimedSDC4



Description: Timed Symbolic Defined Component
Library: Timed, Nonlinear
Class: TSDFTimedSDC4
Derived From: TimedSDC3

Parameters

Name	Description	Default	Unit	Type
FCarrier	Carrier frequency expression, function of inputs' carrier frequencies	0.0	Hz	real
Expression	Expression, function of inputs and "time"	0.0+j*0.0		complex
RIn	input resistance	DefaultRIn	Ohm	real
ROut	output resistance	DefaultROut	Ohm	real
RTemp	physical temperature in degrees C	DefaultRTemp	Celsius	real

Pin Inputs

Pin	Name	Description	Signal Type
1	input1	input signal 1	timed
2	input2	input signal 2	timed
3	input3	input signal 3	timed
4	input4	input signal 4	timed

Pin Outputs

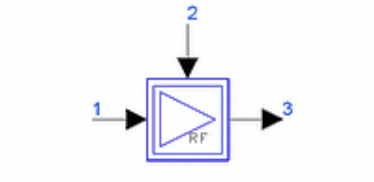
Pin	Name	Description	Signal Type
5	output	output signal	timed

Notes/Equations

1. This component generates a timed data output that is evaluated using an expression. Expression can be any valid expression, following the syntax used for writing expressions on a VAR block.
   VAR exception: The VAR expression must not be based on a Data Access component. To use the Data Access component use one of the numeric SDC components (SDC1, ..., SDC4).
2. Input data is specified by predefined variables _v1, _v2, _fc1, _fc2, etc. where v specifies signal voltages and fc specifies carrier frequencies; 1 and 2 are the port numbers.

3. The Expression can be also be dependent on time and _fc, where time is the simulation time that is incremented by Tstep for each firing of this component decided by the schedule and _fc is the resultant of FCarrier expression.
4. The FCarrier expression can be dependent on _fc1, _fc2, etc.
5. When FCarrier is 0, the imaginary part of the Expression will be ignored.

VcGainRF



Description: Voltage controlled complex gain with gain compression
Library: Timed, Nonlinear
Class: TSDFVcGainRF
Derived From: _GainRF

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
NoiseFigure	input noise figure, in dB	0		real	[0, ∞)
GCType	Gain compression type: none, TOI, dBc1, TOI+dBc1, PSat+GCSat+TOI, PSat+GCSat+dBc1, PSat+GCSat+TOI+dBc1, Gain compression data points vs input power	none		enum	
TOIout	Third order intercept power	3	W	real	(-∞, ∞)†
dBc1out	1 dB gain compression power	1	W	real	(-∞, ∞)†
PSat	Saturation power	1	W	real	(-∞, ∞)†
GCSat	Gain compression at saturation; dB	1		real	[3, 7]†
GComp	Array of triple values for large signal gain change vs signal power. Input Power in dBm, Gain change from small signal in dB, and Phase change from small signal in degree	0 0 0		real array	††
VMin	minimum limit for tune voltage	0	V	real	(-∞, ∞)
VMax	maximum limit for tune voltage	40	V	real	(-∞, ∞)
Min_dB	minimum dB at VMin for linear operation	-10		real	(-∞, ∞)
Max_dB	maximum dB at VMax for linear operation	70		real	(-∞, ∞)
VcType	control voltage to gain characteristic: dB/V, linear	dB/V		enum	

† Values for TOIout, dbc1out, PSat, and GCSat are interdependant as explained in Note 3 of GainRF documentation.
†† Refer to Note 3, section *Gain compression data points vs input power*, of GainRF documentation.

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed
2	control	control signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
3	output	output signal	timed

Notes/Equations

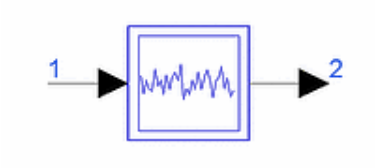
1. VcGainRF is a voltage controlled amplifier model that may include nonlinear gain compression.
2. Refer to GainRF documentation for model details.
3. The small signal gain is controlled by the voltage at pin 2, $V_2(t)$, and the parameters VMin, VMax, Min_dB, Max_dB and VcType. $V_2(t)$ is limited to a lower limit of VMin and a maximum limit of VMax. The minimum gain, Min_dB, occurs for $V_2(t)$=VMin. The maximum gain, Max_dB, occurs at $V_2(t)$=VMax.
4. The gain characteristic for $V_2(t)$ between VMin and VMax is set by VcType to be either a dB/V characteristic or a linear characteristic.
   For VcType=dB/V, the small signal power gain PowerGain(in dB) = Min_dB + ($V_2(t)$-VMin)*(Max_dB-Min_dB)/(VMax-VMin).

   For VcType=linear, the small signal power gain PowerGain(in dB) = 20*log10(VoltageGain); where VoltageGain = ($V_2(t)$-VMin)/(VMax-VMin)*$10^{(Max\_dB/20)}$ + (VMax-$V_2(t)$)/(VMax-VMin)*$10^{(Min\_dB/20)}$
5. For information regarding timed nonlinear component signals, refer to *Timed Nonlinear Components* (timed).

Timed RF Subsystems

- *AddNDensity* (timed)
- *CktAGCLoopFilter* (timed)
- *Ckt MGA 545P8 PA* (timed)
- *Ckt MGA 72543 LNA* (timed)
- *CktRF PA* (timed)
- *CktVI Sense* (timed)
- *PAE Value* (timed)
- *RF Combiner* (timed)
- *RF CombinerIQ* (timed)
- *RF Demod* (timed)
- *RF DemodExtOscRC* (timed)
- *RF DemodFIR* (timed)
- *RF DemodulatorRC* (timed)
- *RF MirrorSignal* (timed)
- *RF Mod* (timed)
- *RF ModFIR* (timed)
- *RF Modulator* (timed)
- *RF ModulatorRC* (timed)
- *RF PAE TestFixture* (timed)
- *RF RX IFout* (timed)
- *RF RX IFout1* (timed)
- *RF RX IFout SBlock* (timed)
- *RF TX IFin* (timed)
- *RF TX IFin1* (timed)
- *RF TX IFin SBlock* (timed)
- *RxAntTempK* (timed)

AddNDensity



Description: Add noise density to input signal
Library: Timed, RF Subsystems
Class: TSDF_AddNDensity
Derived From: baseStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
NDensity	noise spectral density at output, in dBm/Hz	-173.975		real	(-∞, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

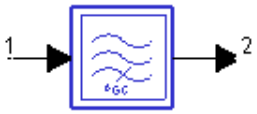
Pin Outputs

Pin	Name	Description	Signal Type
2	output	output signal	timed

Notes/Equations

1. AddNDensity adds white Gaussian noise of the specified noise density to the input signal. Input signal power is preserved when the component is connected to a matched load. Noise power delivered to a matched load is
   NDensity × BW
   where BW is the simulation bandwidth.
   BW = 0.5/TStep for a baseband signal
   BW = 1/TStep for an RF signal
   where TStep is the simulation time step of the input signal
2. Information regarding time domain signal differences between ADS Ptolemy simulations and Circuit Envelope and Transient simulations is given in the *Timed Synchronous Dataflow* (ptolemy) section in the *ADS Ptolemy Simulation* (ptolemy) documentation.

CktAGCLoopFilter



Description: Circuit AGC Loop Filter
Library: Timed, Filters
Class: TSDFCktAGCLoopFilter

Parameters

Name	Description	Default	Sym	Unit	Type	Range
DampingFactor	loop damping factor	0.707			real	(0, 1]
NormalizedZero	loop normalized zero	1			real	
BW	closed loop -3dB frequency	4500	S	Hz	real	(0, ∞)
ExternalGain	open loop gain external to this filter	1000			real	[1, ∞)
C2	loop filter internal capacitor value	0.0000000002		F	real	
OpAmpGain0	loop filter internal opamp gain at DC in dB	120		dB	real	[10, 200]
OpAmpBW	loop filter internal opamp bandwidth	100MHz		Hz	real	(0, ∞)
OpAmpVEE	loop filter internal opamp negative supply voltage	-100		V	real	(-∞, 0)
OpAmpVCC	loop filter internal opamp positive supply voltage	100		V	real	(0, ∞)
MaxTimeStep	circuit transient cosimulation maximum time step value	0.000001		sec	real	(0, 0.1]
RefV	reference voltage for control loop opamp	0.1		V	real	[0, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	In	Loop filter input	timed

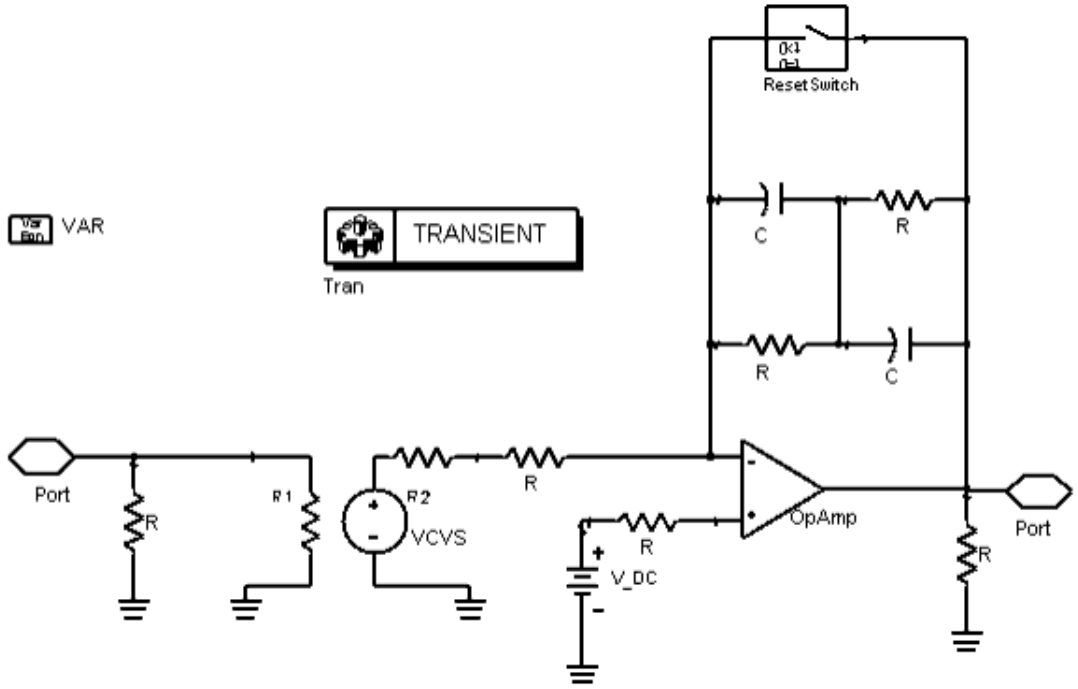
Pin Outputs

Pin	Name	Description	Signal Type
2	Out	Loop filter output	timed

Notes/Equations

1. This component provides a control loop filter designed for use with RF AGC loops. It is a key component for RF AGC loop simulation.
2. This is a single-rate component. Each firing, one input token is consumed and one output token is produced.
3. The structure of this component is shown below.

CktAGCLoopFilter Structure

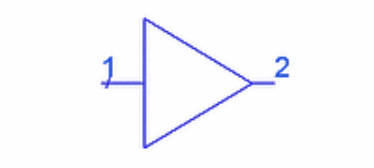


4. A voltage-controlled voltage source followed by an OpAmp component forms the loop filter to work with the AGC loop.
5. A second-order feedback loop is formed based on the DampingFactor and NormalizedZero parameters.
6. ExternalGain specifies gain external to this filter; internal filter gain will be automatically set to achieve the specified second-order loop characteristics.
7. OpAmpGain0, OpAmpBW, OpAmpVEE, and OpAmpVCC parameters set the appropriate OpAmp working points.
8. MaxTimeStep is used by the nested circuit transient cosimulation.
9. RefV specifies the OpAmp reference voltage and stabilizes the control loop operation.

References

1. M. Jeruchim, P. Balaban and K. Shanmugan, "Simulation of Communication System," Plenum Press, New York and London, 1992.

Ckt_MGA_545P8_PA



Description: Medium power GaAs RFIC amplifier circuit model from Avago Technologies
Library: Timed, RF Subsystems
Class: TSDFCkt_MGA_545P8_PA

Parameters

Name	Description	Default	Unit	Type	Range
CE_TimeStep	Circuit envelope simulation time step	12.5 nsec	sec	real	(0, +∞)
FCarrier	Input RF signal carrier frequency	5.2 GHz	Hz	real	(0, +∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	RF_in	RF input signal	timed

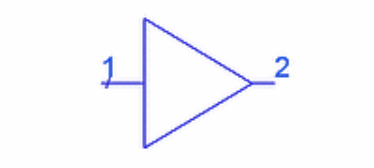
Pin Outputs

Pin	Name	Description	Signal Type
2	RF_out	RF output signal	timed

Notes/Equations

1. Ckt_MGS_545P8_PA is an Analog/RF Schematic design and is a medium power nonlinear RF amplifier with two FET transistor stages. The amplifier model is for the Avago Technologies MGA-545P8 RFIC with low current and +22 dBm output power designed for 5-6 GHz systems.
2. Pin 1 is the RF input; pin 2 is the RF output.
3. This amplifier has no default artwork associated with it.
4. Use of this component is demonstrated in the *File* > *Open* > *Example* > PtolemyDocExamples > Timed_RF_Subsystems_wrk_. Open the networks design *MGA_545P8*<design_name>*, where _<design_name>* may be *SParam, HBtest_P1tone, HBtest_P2tones, HBtest_FSweep_P1tone* or *WLAN_802_11a_EVM*.
5. Information regarding the Avago Technologies MGA-545P8 RFIC is available at the following Avago web pages:
   Data sheet: http://www.avagotech.com/docs/5989-1810EN
   Application note: http://www.avagotech.com/docs/5989-0397EN

Ckt_MGA_72543_LNA



Description: Low noise PHEMT RFIC amplifier circuit model from Avago Technologies
Library: Timed, RF Subsystems
Class: TSDFCkt_MGA_72543_LNA

Parameters

Name	Description	Default	Unit	Type	Range
CE_TimeStep	Circuit envelope simulation time step	1/(3.84e6)/8	sec	real	(0, +∞)
FCarrier	Input RF signal carrier frequency	1900 MHz	Hz	real	(0, +∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	RF_in	RF input signal	timed

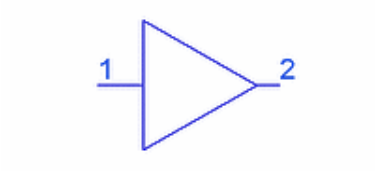
Pin Outputs

Pin	Name	Description	Signal Type
2	RF_out	RF output signal	timed

Notes/Equations

1. Ckt_MGA_72543_LNA is an Analog/RF Schematic design that is a PHEMT low noise amplifier (LNA). The amplifier model is for the Avago Technologies MGA-72543 RFIC with 3V dc bias and adjustable IIP3 from +2 to +14 dBm for 0.1 to 6.0 GHz operation.
2. Pin 1 is the RF input; pin 2 is the RF output.
3. This amplifier has no default artwork associated with it.
4. Use of this component is demonstrated in the *File* > *Open* > *Example* > PtolemyDocExamples > Timed_RF_Subsystems_wrk_. Open the networks design *MGA_72543*<design_name>*, where _<design_name>* may be *SParam, HBtest_P1tone, HBtest_P2tones, HBtest_FSweep_P1tone or TDSCDMA_UpLnk_Power*.
5. Information regarding the Avago Technologies MGA-72543 RFIC is available at the following Avago web pages:
   Data sheet: http://www.avagotech.com/docs/AV02-1296EN
   Application note: http://www.avagotech.com/docs/MPUB-3754

CktRF_PA



Description: RF power amplifier, analog circuit
Library: Timed, RF Subsystems
Class: TSDFCktRF_PA

Parameters

Name	Description	Default	Unit	Type	Range
TStep	system level simulation time step	0.001	sec	real	(0, ∞)
FCarrier	input signal RF carrier frequency	1000000	Hz	real	(0, ∞)
Bias1	DC bias voltage 1	-0.8005247	V	real	(-∞, ∞)
Bias2	DC bias voltage 2	4.602021	V	real	
Bias3	DC bias voltage 3	-0.8008529	V	real	
Bias4	DC bias voltage 4	5.492358	V	real	
Bias5	DC bias voltage 5	0	V	real	
Bias6	DC bias voltage 6	0	V	real	

Pin Inputs

Pin	Name	Description	Signal Type
1	RFin	RF input	timed

Pin Outputs

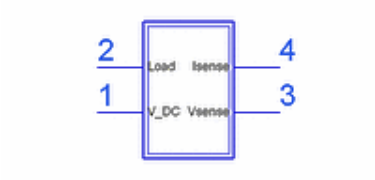
Pin	Name	Description	Signal Type
2	RFout	RF output	timed

Notes/Equations

1. CktRF_PA is an Analog/RF Schematic design and is a nonlinear RF amplifier with two FET transistor stages and four bias voltages. The amplifier delivers approximately 30dB gain with 15 dBm saturated power with a 2.5 dB noise figure.
2. Pin 1 is the RF input; pin 2 is the RF output.
3. Inside the CktRF_PA design, there are internal global named nodes for sensed dc bias voltage and current named Bias1_V, Bias1_I, ... , Bias6_V, Bias6_I. Only Bias1 through Bias4 are used for this design. These sensed values can be used outside this design to enable calculation of amplifier power added efficiency (PAE). For details regarding PAE measurement, refer to RF_PAE_TestFixture.
4. This amplifier has no default artwork associated with it.
5. Use of this component is demonstrated in the *File* > *Open* > *Example* > PtolemyDocExamples > Timed_RF_Subsystems_wrk_. Open the networks design *RF_PAE_example* and push into the RF_PAE_TestFixture where CktRF_PA is used.

6. Information regarding time domain signal differences between ADS Ptolemy simulations and Circuit Envelope and Transient simulations is given in the *Timed Synchronous Dataflow* (ptolemy) section in the *ADS Ptolemy Simulation* (ptolemy) documentation.

CktVI_Sense



Description: Voltage and current sense network, analog circuit
Library: Timed, RF Subsystems
Class: TSDFCktVI_Sense

Pin Inputs

Pin	Name	Description	Signal Type
1	V1	Signal input	timed

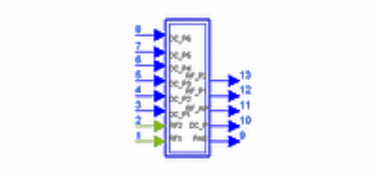
Pin Outputs

Pin	Name	Description	Signal Type
2	V2	Signal output	timed
3	VSense	sense voltage output	timed
4	ISense	sense current output	timed

Notes/Equations

1. CktVI_Sense is an Analog/RF Schematic design and is a purely linear, dependent source model. Nonlinear controlled sources are available in the Nonlinear Devices library.
2. This source is assumed to be noiseless.
3. Pin 1 should be connected to the DC source. Pin 2 should be used as the output of the DC source to which the circuit is connected.
4. Pins 3 and 4 are isolated from pins 1 and 2. Pin 3 is the detected voltage of the DC source. Pin 4 is the detected current flowing from pin 1 to pin 2 and is represented as a voltage value with 1V/A.
5. This source has no default artwork associated with it.
6. Use of this component is demonstrated in the *File* > *Open* > *Example* > PtolemyDocExamples > Timed_RF_Subsystems_wrk_. Open the networks design *RF_PAE_example* and push into RF_PAE_TestFixture then CktRF_PA. CktRF_PA includes an application of four instances of CktVI_Sense to sense the voltage and current of the four V_DC sources.
7. Information regarding time domain signal differences between ADS Ptolemy simulations and Circuit Envelope and Transient simulations is given in the *Timed Synchronous Dataflow* (ptolemy) section in the *ADS Ptolemy Simulation* (ptolemy) documentation.

PAE_Value



Description: Calculates PAE from RF and DC input values
Library: Timed, RF Subsystems
Class: TSDFPAE_Value

Parameters

Name	Description	Default	Unit	Type	Range
NumDC	number of DC power values used	0.001		int	[1, +6)
NumStart	sample number to start integration for power in watts	0		int	[0, ∞)
RefR	RF signal reference resistance	50	Ohm	real	(0, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	RF1	RF 1 envelope input	complex
2	RF2	RF 2 envelope input	complex
3	DC_P1	DC 1 bias power	real
4	DC_P2	DC 2 bias power	real
5	DC_P3	DC 3 bias power	real
6	DC_P4	DC 4 bias power	real
7	DC_P5	DC 5 bias power	real
8	DC_P6	DC 6 bias power	real

Pin Outputs

Pin	Name	Description	Signal Type
9	PAE	power added efficiency for PA tested	real
10	DC_Power	dc power (Watts) consumed by PA tested	real
11	RF_AddedPower	RF power added (Watts) by PA tested	real
12	RF_P1	RF power (Watts) for RF 1 signal	real
13	RF_P2	RF power (Watts) for RF 2 signal	real

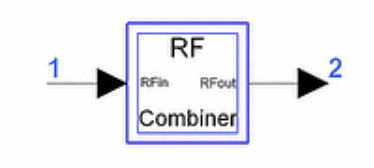
Notes/Equations

1. PAE_Value is a hierarchical model composed of other components. It is used to calculate power-added-efficiency (PAE) and associated values from input signal representing analog/RF circuit dc bias power levels and RF input and output signal envelopes.
2. This model is used inside the RF_PAE_TestFixture model; for details, refer to RF_PAE_TestFixture.
3. Use of this component is demonstrated in the *File* > *Open* > *Example* >

PtolemyDocExamples > Timed_RF_Subsystems_wrk_. Open the networks design *RF_PAE_example* where RF_PAE_TestFixture is used.

4. Information regarding time domain signal differences between ADS Ptolemy simulations and Circuit Envelope and Transient simulations is given in the *Timed Synchronous Dataflow* (ptolemy) section in the *ADS Ptolemy Simulation* (ptolemy) documentation.

RF_Combiner



Description: RF combiner, combining desired signal with undesired CW and modulated interferers
Library: Timed, RF Subsystems
Class: TSDFRF_Combiner
Derived From: baseSubcircuit

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
Gain	power gain applied to input signal, in dB	-100		real	(-∞, ∞)
SymbolRate	symbol rate of input and modulated interferer signals	25e3	Hz	real	(0, ∞)
SampPerSym	samples per symbol period	4		int	(0, ∞)
InputFreq	input signal frequency	900e6	Hz	real	(0, ∞)
CWInterfererFreq	CW interferer frequency	900.1e6	Hz	real	(0, ∞)
CWInterfererPower	CW interferer power	1e-6	W	real	[0, ∞)
ModInterfererFreq	modulated interferer frequency	900.02e6	Hz	real	(0, ∞)
ModInterfererPower	modulated interferer power	1e-6	W	real	[0, ∞)
ModInterfererExcessBW	excess bandwidth of modulated interferer raised cosine filters	0.35		real	[0, 1]
AutoResampling	resample input signal to support bandwidth of combined signal? NO, YES	YES		enum	

Pin Inputs

Pin	Name	Description	Signal Type
1	RFin	input signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
2	RFout	output signal	timed

Notes/Equations

1. RF_Combiner is a hierarchical model composed of other components. It can be used to attenuate (or amplify) the input signal and add interference to it. Both a CW and a modulated interferer are available. For each input sample consumed, N output samples are produced, where N is the internal upsample ratio discussed in Note 3.
2. The Gain parameter can be used to set the power gain (in dB) for the input signal.

Positive values will amplify the input signal, whereas negative values will attenuate it.

3. To add the interferers to the input signal without any loss of information (aliasing) the input signal may need to be upsampled. The upsampling ratio is such that the resulting simulation TStep satisfies the following inequality:

$$TStep \le \frac{0.2}{f_{max} - f_{min}}$$

where f_{max} (f_{min}) is the maximum (minimum) of InputFreq, CWInterfererFreq, and ModInterfererFreq.
For example, assume
SymbolRate = 1 MHz
SampPerSym = 8
InputFreq = 2 GHz
CWInterfererFreq = 1.995 GHz
ModInterfererFreq = 2.005 GHz
The input signal TStep is

$$\frac{1}{8}\ \mu\text{sec} = 125\ \text{nsec}$$

The maximum TStep that will not cause loss of information is

$$\frac{0.2}{10\ \text{MHz}} = 20\ \text{nsec}$$

Therefore, the upsampling ratio is set to

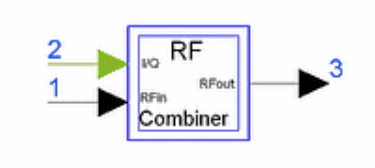
$$int\frac{125}{20} + 1 = 7$$

There are two ways the user can determine the internal upsampling ratio of the RF_Combiner:

- do the math described by the equations in the example above
- export the value of the upsampling ratio to the data display using the DF controller Output tab. For example, assume that the RF_Combiner is used on the top-level schematic and its instance name is R1. Then after clicking the Add/Remove button (of the Output tab in the DF controller), click on the cross next to R1, select Ratio, and click on the Add button.
  The *instance_name.Ratio* notation can be used anywhere (in equations, as a component parameter value) on the level of hierarchy that uses an RF_Combiner component to reference its internal upsampling ratio.
  If this automatic resampling of the input signal is not desired, then the AutoResampling parameter should be set to NO.

4. The CW interferer frequency and power can be set using the parameters CWInterfererFreq and CWInterfererPower.
5. The modulated interferer signal is a QPSK signal with symbol rate equal to SymbolRate and filtered with a square root raised-cosine filter of excess bandwidth equal to ModInterfererExcessBW. Therefore, the bandwidth of the modulated interferer signal is SymbolRate × (1+ModInterfererExcessBW)/2.
6. Information regarding time domain signal differences between ADS Ptolemy simulations and Circuit Envelope and Transient simulations is given in the *Timed Synchronous Dataflow* (ptolemy) section in the *ADS Ptolemy Simulation* (ptolemy) documentation.

RF_CombinerIQ



Description: RF combiner with IQ input combining desired signal with undesired CW and modulated interferers
Library: Timed, RF Subsystems
Class: TSDFRF_CombinerIQ
Derived From: baseSubcircuit

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
Gain	Power gain applied to input signal (dB)	-100		real	(-∞, ∞)
SymbolRate	Symbol rate of input and modulated interferer signals	25e3	Hz	real	(0, ∞)
SampPerSym	Samples per symbol period	4		int	(0, ∞)
InputFreq	Input signal frequency	900e6	Hz	real	(0, ∞)
CWFreq	CW interferer frequency	900.1e6	Hz	real	(0, ∞)
CWPower	CW interferer power	1e-6	W	real	[0, ∞)
ModFreq	Modulated interferer frequency	900.02e6	Hz	real	(0, ∞)
ModPower	Modulated interferer power	1e-6	W	real	[0, ∞)
ModVRef	Modulated interferer reference voltage for output power calibration	1.0		real	(0, ∞)
ModFilterLength	Modulated interferer raised cosine filter length (symbols)	16		int	[1, ∞)
ModSquareRoot	Modulated interferer square root raised cosine? NO, YES	YES		enum	
ModExcessBW	Modulated interferer raised cosine filter excess bandwidth	0.35		real	[0, 1]
AutoResampling	Resample input signal to support bandwidth of combined signal? NO, YES	YES		enum	

Pin Inputs

Pin	Name	Description	Signal Type
1	RFin	input signal	timed
2	IQ	IQ interferer signal	complex

Pin Outputs

Pin	Name	Description	Signal Type
3	RFout	output signal	timed

Notes/Equations

1. RF_CombinerIQ is a hierarchical model composed of other components. It can be

used to attenuate (or amplify) the input signal and add interference to it. Both a CW and a modulated interferer are available. For each input sample consumed, N output samples are produced, where N is the internal upsample ratio discussed in Note 3.

2. The Gain parameter can be used to set the power gain (in dB) for the input signal. Positive values will amplify the input signal, whereas negative values will attenuate it.
3. To add the interferers to the input signal without any loss of information (aliasing) the input signal may need to be upsampled. The upsampling ratio is such that the resulting simulation TStep satisfies the following inequality:

$$TStep \le \frac{0.2}{f_{max} - f_{min}}$$

where f_{max} (f_{min}) is the maximum (minimum) of InputFreq, CWFreq, and ModFreq.

For example, assume
SymbolRate = 1 MHz
SampPerSym = 8
InputFreq = 2 GHz
CWFreq = 1.995 GHz
ModFreq = 2.005 GHz
The input signal TStep is

$$\frac{1}{8}\,\mu\text{sec} = 125\ \text{nsec}$$

The maximum TStep that will not cause loss of information is

$$\frac{0.2}{10\ \text{MHz}} = 20\ \text{nsec}$$

Therefore, the upsampling ratio is set to

$$int\frac{125}{20} + 1 = 7$$

There are two ways the user can determine the internal upsampling ratio of the RF_CombinerIQ:

- do the math described by the equations in the example above
- export the value of the upsampling ratio to the data display using the DF controller Output tab. For example, assume that the RF_CombinerIQ is used on the top-level schematic and its instance name is R1. Then after clicking the Add/Remove button (of the Output tab in the DF controller), click on the cross next to R1, select Ratio, and click on the Add button.
  The *instance_name.Ratio* notation can be used anywhere (in equations, as a component parameter value) on the level of hierarchy that uses an RF_CombinerIQ component to reference its internal upsampling ratio.
  If this automatic resampling of the input signal is not desired, then the AutoResampling parameter should be set to NO.

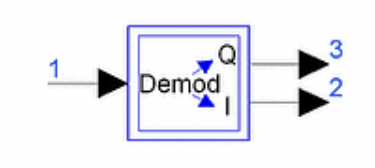
4. The CW interferer frequency and power can be set using the parameters CWFreq and CWPower.
5. The modulated interferer signal is provided as a numeric complex signal to the second input of RF_CombinerIQ. This signal is assumed to be the baseband I and Q envelopes of the interferer signal with one sample per symbol. The baseband I and Q envelopes are upsampled, filtered, and used to modulate the in-phase and quadrature phase carriers of a QAM modulator. The filter used is a raised cosine filter whose length, excess bandwidth, and type (raised cosine or root raised cosine) can be set using ModFilterLength, ModExcessBW, and ModSquareRoot parameters, respectively.
   The modulated interferer frequency and power can be set using the parameters ModFreq and ModPower. To get the correct modulated interferer power the ModVRef

parameter must be set to the rms value of the baseband I and Q interferer envelopes. If this rms value is not known a CxToRect converter followed by the TkIQrms component can be connected to the interferer signal input and the rms value will be displayed in the "Agilent Control Panel" window that will pop up after the simulation starts.

The symbol rate for the modulated interferer signal is the same as the symbol rate of the input signal (given by the parameter SymbolRate). Therefore, the bandwidth of the modulated interferer signal is SymbolRate × (1+ModExcessBW)/2.

6. Information regarding time domain signal differences between ADS Ptolemy simulations and Circuit Envelope and Transient simulations is given in the *Timed Synchronous Dataflow* (ptolemy) section in the *ADS Ptolemy Simulation* (ptolemy) documentation.

RF_Demod



Description: RF demodulator with I/Q output
Library: Timed, RF Subsystems
Class: TSDFRF_Demod
Derived From: baseSubcircuit

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
SymbolRate	symbol rate	25e3	Hz	real	(0, ∞)
ExcessBW	raised cosine filter excess bandwidth	0.35		real	[0, 1]
SquareRoot	use square root raised cosine model: No, Yes	Yes		enum	
FCarrier	carrier frequency	10e6	Hz	real	{-1} or (0, ∞)
Phase	reference phase in degrees	0		real	(-∞, ∞)
VRef	reference voltage for output calibration	1.0	V	real	(0, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	RF_in	input RF signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
2	I_out	I output	timed
3	Q_out	Q output	timed

Notes/Equations

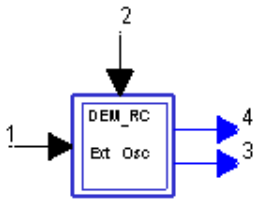
1. RF_Demod is a hierarchical model composed of other components. The input to the demodulator is an RF signal. The input signal is demodulated and then its in-phase and quadrature phase components are filtered. For each input sample consumed, one output sample is produced.
2. The FCarrier parameter sets the internal oscillator frequency used for demodulation. Setting FCarrier to -1 will use the input signal characterization frequency as the internal oscillator frequency.
3. RF_Demod is calibrated so that its sampled (at the optimal sampling instants) output values are the same as the values at the input of RF_Mod when the following conditions are satisfied:
    - both the modulator and demodulator use square root raised-cosine filtering with

the same excess bandwidth

- power at the demodulator input is 10 mW = 10 dBm
- VRef is set to the same value for the modulator and the demodulator
  If the demodulator input power is different from 10 mW then its VRef parameter must be set appropriately to compensate for that. Let R equal the ratio of 10 mW to the actual input power of the demodulator. Then the demodulator's VRef must be set to the VRef value of the modulator multiplied by
  $\sqrt{R}$
  For example, assume that the demodulator input power is 40 mW and the VRef parameter of the modulator is 2. Then the demodulator VRef must be set to
  $$2 \times \sqrt{\frac{10}{40}} = 2 \times \frac{1}{2} = 1$$

4. Information regarding time domain signal differences between ADS Ptolemy simulations, Circuit Envelope, and Transient simulations is given in the *Timed Synchronous Dataflow* (ptolemy) section in the *ADS Ptolemy Simulation* (ptolemy) documentation.

RF_DemodExtOscRC



Description: RF demodulator with External Oscilator
Library: Timed, RF Subsystems
Class: TSDFRF_DemodExtOscRC

Parameters

Name	Description	Default	Unit	Type	Range
RIn	Input resistance	50 Ohm	Ohm	real	(0, ∞)
RTemp	Physical temperature	-273.15	Celsius	real	[-273.15, ∞)
MirrorSpectrum	Mirror spectrum about carrier? NO, YES	NO		enum	
GainImbalance	Gain imbalance in dB, Q vs I	0	dB	real	(-∞, ∞)
PhaseImbalance	Phase imbalance in degrees, Q vs I	0	deg	real	(-∞, ∞)
Sensitivity	Voltage output sensitivity, Vout/Vin	1		real	(-∞, ∞)
SymbolRate	Symbol rate	3.84 MHz	Hz	real	(0, ∞)
SampPerSym	Sample per symbol	8		int	[1, ∞)
FilterLength	Filter length	16		int	[1, ∞)
ExcessBW	Excess bandwidth for RC filter	0.5		real	[0, 1]
SquareRoot	Squareroot ? NO, YES	YES		enum	

Pin Inputs

Pin	Name	Description	Signal Type
1	RF	Input I	timed
2	OSC	Input Q	timed

Pin Outpus

Pin	Name	Description	Signal Type
3	I_Out	Output I	timed
4	Q_Out	Output Q	timed

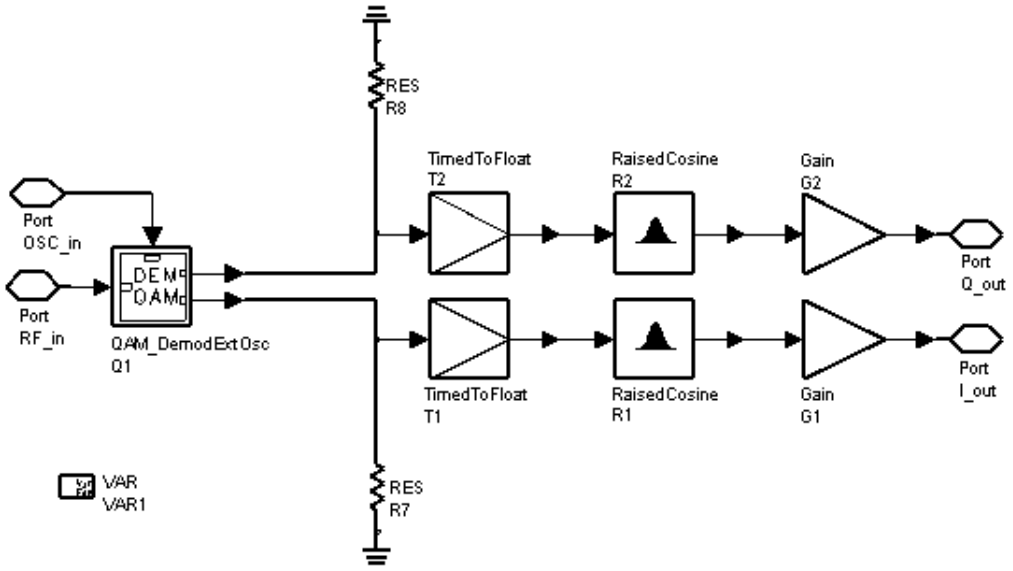
Notes/Equations

1. This component is used for demodulation of RF signals with external oscillation. It is key for RF PLL simulation where local oscillator control for detecting phase/frequency errors in the feedback loop is needed.
   The schematic for this network is shown in RF_DemoExtOscRC Schematic.
2. This is a single-rate component. Each firing, one input token is consumed for both input pins and one token is produced for both output pins.
   The input RF pin must be connected to the reference signal and input OSC pin must

be connected to the test signal.

3. Timed input RF and external oscillator signals are connected to QAM_DemodExtOsc. Demodulated output timed signals are then converted to an I,Q signal by two TimedToFloat components and sent to two RaisedCosine filters, each followed by a Gain component. Demodulated I and Q data is sent to the I_Out and Q_Out pins, respectively.
4. The GainImbalance and PhaseImbalance parameters set the gain and phase imbalance distortions introduced by the demodulator.
5. The SampPerSym, FilterLength, ExcessBW, and SquareRoot parameters set the properties of the raised cosine filters used to filter the demodulated I and Q envelopes.
6. The MirrorSpectrum parameter can be used to mirror the signal about its carrier frequency. This is useful if the configuration of the mixers in the transmitter and receiver has resulted in a signal that is mirrored (conjugated) with respect to the signal at the input of the modulator.

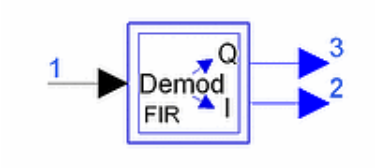
RF_DemoExtOscRC Schematic



References

1. M. Jeruchim, P. Balaban and K. Shanmugan, "Simulation of Communication System," Plenum Press, New York and London, 1992.

RF_DemodFIR



Description: RF demodulator with user defined FIR filter taps
Library: Timed, RF Subsystems
Class: TSDFRF_DemodFIR
Derived From: baseSubcircuit

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
SampPerSym	samples per symbol period	8		int	[1, ∞)
FilterTaps	FIR filter taps	< $HPEESOF_DIR/adsptolemy/lib/data /square_root_raised_cosine.txt		real array	See Note 3
FCarrier	carrier frequency	2e9	Hz	real	{-1} or (0, ∞)
Phase	reference phase in degrees	0	deg	real	(-∞, ∞)
VRef	reference voltage for output calibration	1	V	real	(0, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	RF_in	input RF signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
2	I_out	I output	real
3	Q_out	Q output	real

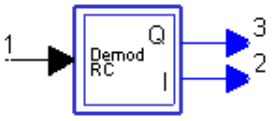
Notes/Equations

1. RF_DemodFIR is a hierarchical model composed of other components. The input to the demodulator is an RF signal. The input signal is demodulated and then its in-phase and quadrature phase components are filtered. For each input sample consumed, one output sample is produced.
2. The FCarrier parameter sets the internal oscillator frequency used for demodulation. Setting FCarrier to -1 will use the input signal characterization frequency as the internal oscillator frequency.
3. The FilterTaps parameter is used to set the coefficients of the FIR filter that filters the demodulated I and Q signals. The filter coefficients must be normalized so that the

sum of their squares equals 1. If this normalization is not done then the signal levels at the output of the component cannot be calibrated as explained in note 4. The default value for this parameter corresponds to a square root raised-cosine filter with 65 taps, 8 samples per symbol and 0.5 excess bandwidth.

4. RF_DemodFIR is calibrated so that its sampled (at the optimal sampling instants) output values are the same as the values at the input of the RF_ModFIR when the following conditions are satisfied:
   - both the modulator and demodulator use square root raised-cosine filtering with the same excess bandwidth and filter coefficients normalized so that the sum of their squares equals 1.
   - power at the demodulator input is 10 mW = 10 dBm
   - VRef is set to the same value for both the modulator and the demodulator
     If the demodulator input power is different from 10 mW then its VRef parameter must be set appropriately to compensate for that. Let R equal the ratio of 10 mW to the actual input power of the demodulator. Then the demodulator's VRef must be set to the VRef value of the modulator multiplied by
     $\sqrt{R}$
     For example, assume that the demodulator input power is 40 mW and the VRef parameter of the modulator is 2. Then the demodulator VRef must be set to
     $$2 \times \sqrt{\frac{10}{40}} = 2 \times \frac{1}{2} = 1$$
5. If the filters used in the modulator and demodulator are not square root raised-cosine then the signals at the output of the demodulator will have ISI. In this case, there is no sampling instant where the signals can be sampled that will recover the exact same values as the ones at the input of the modulator. Nevertheless, if the filter coefficient normalization is done for both the modulator and demodulator filters and VRef is set as explained in note 4 then the rms value of the signal at the output of the demodulator (sampled at the points where ISI is minimum) will be equal to the rms value of the signal at the input of the modulator.
6. Information regarding time domain signal differences between ADS Ptolemy simulations and Circuit Envelope and Transient simulations is given in the *Timed Synchronous Dataflow* (ptolemy) section in the *ADS Ptolemy Simulation* (ptolemy) documentation.

RF_DemodulatorRC



Description: RF demodulator with raised cosine filters
Library: Timed, RF Subsystems
Class: TSDFRF_DemodulatorRC
Derived From: baseSubcircuit

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
RTemp	physical temperature, in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
FCarrier	Demodulator internal oscillator carrier frequency	1900 MHz	Hz	real	{-1} or (0, ∞)†
Phase	Demodulator reference phase	0.0	deg	real	(-∞, ∞)
VRef	Reference voltage for output calibration	1.0 V	V	real	(0, ∞)
SampPerSym	Samples per symbol period	8		int	[1, ∞)
FilterLength	Length of raised cosine filters in number of symbols	16		int	[1, ∞)
ExcessBW	Excess bandwidth of raised cosine filters	0.35		real	[newpro:0, 1]
SquareRoot	Square root raised cosine pulse? NO, YES	YES		enum	
MirrorSpectrum	Mirror spectrum around about carrier? NO, YES	NO		enum	
GainImbalance	Gain imbalance in dB, Q channel relative to I channel	0.0	dB	real	(-∞, ∞)
PhaseImbalance	Phase imbalance in degrees, Q channel relative to I channel	0.0	deg	real	(-∞, ∞)

† When set to -1 the carrier frequency of the input signal will be used.

Pin Inputs

Pin	Name	Description	Signal Type
1	RF_in	input RF signal	timed

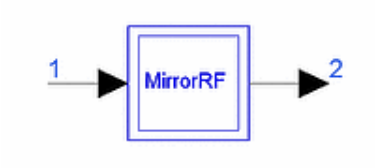
Pin Outputs

Pin	Name	Description	Signal Type
2	I_out	I output	real
3	Q_out	Q output	real

Notes/Equations

1. This component is used to demodulate an RF input signal. The demodulated I and Q envelopes are then filtered with raised cosine filters. For each input sample consumed, one output sample is produced.
2. The FCarrier parameter sets the carrier frequency of the internal oscillator used for the demodulation. If FCarrier is set to -1, then the input signal characterization frequency is used.
3. The Phase parameter sets the reference phase for the internal oscillator used for the demodulation.
4. The GainImbalance and PhaseImbalance parameters set the gain and phase imbalance distortions introduced by the demodulator.
5. The SampPerSym, FilterLength, ExcessBW, and SquareRoot parameters set the properties of the raised cosine filters used to filter the demodulated I and Q envelopes.
6. The MirrorSpectrum parameter can be used to mirror the signal about its carrier frequency. This is useful if the configuration of the mixers in the transmitter and receiver has resulted in a signal that is mirrored (conjugated) with respect to the signal at the input of the modulator.
7. RF_DemodulatorRC is calibrated so that its sampled (at the optimal sampling instants) output values are the same as the values at the input of the RF_ModulatorRC component when the following conditions are satisfied:
   - both the modulator and demodulator use square root raised-cosine filtering with the same excess bandwidth
   - power at the demodulator input is 10 mW = 10 dBm
   - VRef is set to the same value for the modulator and the demodulator
     If the demodulator input power is different from 10 mW then its VRef parameter must be set appropriately to compensate for that. Let R equal the ratio of 10 mW to the actual input power of the demodulator. Then the demodulator's VRef must be set to the VRef value of the modulator multiplied by
     $\sqrt{R}$
     For example, assume that the demodulator input power is 40 mW and the VRef parameter of the modulator is 2. Then the demodulator VRef must be set to
     $$2 \times \sqrt{\frac{10}{40}} = 2 \times \frac{1}{2} = 1$$
8. If this component is connected to the output of RF_ModulatorRC or if there are components between them that do not change the rate (upsampling or downsampling) then the SampPerSym parameter of both components must be the same.
   If there are components between RF_ModulatorRC and RF_DemodulatorRC that change the rate then the user must determine (based on the value of the SampPerSym parameter of the RF_ModulatorRC and the upsampling/downsampling ratios of the intermediate components) how many samples per symbol there are at the input of RF_DemodulatorRC. The SampPerSym parameter of the RF_DemodulatorRC must be set to this number, which must be an integer.

RF_MirrorSignal



Description: RF mirror signal
Library: Timed, RF Subsystems
Class: TSDF_RF_MirrorSignal
Derived From: baseStar

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
MirrorSignal	Mirror Signal?: NO, YES	NO		enum	

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input RF signal	timed

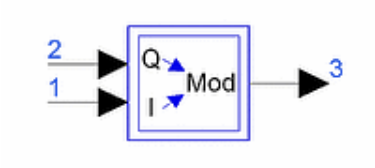
Pin Outputs

Pin	Name	Description	Signal Type
2	output	output RF signal	timed

Notes/Equations

1. This component mirrors the input RF signal when the MirrorSignal parameter is set to YES; if the MirrorSignal parameter is set to NO then the output signal is the same as the input signal. The input signal must be in a complex envelope representation in order for any mirroring to occur. If the input signal is in a baseband representation, mirroring will not occur even if MirrorSignal is set to YES. For mirroring to occur if the input signal is an RF signal in a baseband representation, use the FcChange component before and after RF_MirrorSignal to change the signal representation to a complex envelope and back to a baseband representation.
2. In the time domain, a mirrored signal has its quadrature phase envelope inverted (compared to a non-mirrored signal). In the frequency domain, a mirrored signal has its spectrum flipped around the characterization frequency (compared to a non-mirrored signal).
3. Information regarding time domain signal differences between ADS Ptolemy simulations and Circuit Envelope and Transient simulations is given in the *Timed Synchronous Dataflow* (ptolemy) section in the *ADS Ptolemy Simulation* (ptolemy) documentation.

RF_Mod



Description: RF modulator with I/Q input
Library: Timed, RF Subsystems
Class: TSDFRF_Mod
Derived From: baseSubcircuit

Parameters

Name	Description	Default	Sym	Unit	Type	Range
RIn	input resistance	DefaultRIn		Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut		Ohm	real	(0, ∞)
FCarrier	carrier frequency	10e6		Hz	real	(0, ∞)
Power	modulator output power	1e-3	P	W	real	[0, ∞)
VRef	reference voltage for output power calibration	1.0		V	real	(0, ∞)
SymbolRate	symbol rate	25e3		Hz	real	(0, ∞)
ExcessBW	raised cosine filter excess bandwidth	0.35			real	[0, 1]
SquareRoot	use square root raised cosine model: No, Yes	Yes			enum	
PhasePolarity	if set to Invert, Q channel signal is inverted: Normal, Invert	Normal			enum	
GainImbalance	gain imbalance in dB, Q channel relative to I channel	0.0			real	(-∞, ∞)
PhaseImbalance	phase imbalance in degrees, Q channel relative to I channel	0.0		deg	real	(-∞, ∞)
I_OriginOffset	I origin offset in percent with repect to output rms voltage	0.0			real	(-∞, ∞)
Q_OriginOffset	Q origin offset in percent with repect to output rms voltage	0.0			real	(-∞, ∞)
IQ_Rotation	IQ rotation, in degrees	0.0		deg	real	(-∞, ∞)
NDensity	noise spectral density at output, in dBm/Hz	-173.975			real	(-∞, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	I_in	I input	timed
2	Q_in	Q input	timed

Pin Outputs

Pin	Name	Description	Signal Type
3	RF_out	output RF signal	timed

Notes/Equations

1. RF_Mod is a hierarchical model composed of other components. Inputs are the baseband I and Q signals that are filtered and used to modulate the in-phase and quadrature phase carriers of a QAM modulator. For each input sample consumed, one output sample is produced.
2. This component assumes that the input I and Q signals are multi-level NRZ waveforms with multiple samples per symbol and that the signal level during a symbol period is constant. Therefore, the raised-cosine or root-raised-cosine filters used to filter the I and Q input signals have an f/sin(f) shaped equalization to compensate for the finite width of the input NRZ waveforms.
   If the input I and Q signals have only 1 sample per symbol then the RF_ModFIR component must be used.
3. The VRef parameter is used to calibrate the modulator. VRef is the input voltage value that results in an instantaneous output power on a matched load equal to P. In order to get an average output power on a matched load equal to P, the input rms voltage must equal VRef. Therefore, to calibrate the modulator, VRef must be set to the input rms voltage.

   Note
   If you are having difficulty calibrating the output power of RF_Mod, push into it and follow the instructions shown on the schematic.
4. The PhasePolarity parameter can be used to invert the polarity of the Q channel signal before modulation. Depending on the configuration and number of mixers in the transmitter and receiver, the output of the demodulator may be inverted. If such a configuration is used, the Q channel signal can be correctly recovered by setting this parameter to Invert.
5. The I_OriginOffset, Q_OriginOffset, IQ_Rotation, GainImbalance, PhaseImbalance, and NDensity parameters are used to add certain impairments to the ideal transmitted signal. Impairments are added in the order described here.
   The filtered I and Q baseband input signals are applied to the I and Q inputs of a QAM modulator, which will apply the gain and phase imbalance to its quadrature phase input. The signal at the output of the QAM modulator is given by:

   $$V_3(t) = A\left(V_1(t)\cos(\omega_c t) - gV_2(t)\sin\left(\omega_c t + \frac{\phi\pi}{180}\right)\right)$$

   where A is a scaling factor that depends on the Power, VRef and ROut parameters specified by the user, $V_1(t)$ is the in-phase input, $V_2(t)$ is the quadrature phase input, g is the gain imbalance

   $$g = 10^{\frac{GainImbalance}{20}}$$

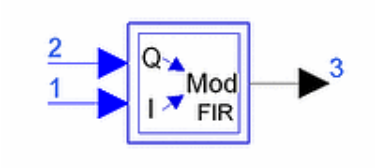
   and, φ (in degrees) is the phase imbalance.
   Next, the signal $V_3(t)$ is rotated by IQ_Rotation degrees. The I_OriginOffset and Q_OriginOffset are then applied to the rotated signal. Note that the amounts specified are percentages with respect to the output rms voltage. The output rms voltage is given by

   $$\sqrt{2 \cdot ROut \cdot P}$$

   Finally, additive noise of spectral density NDensity dBm/Hz is added to the signal. If it is necessary to generate an ideal signal I_OriginOffset, Q_OriginOffset, IQ_Rotation, GainImbalance and PhaseImbalance must all be set to zero, whereas NDensity must be set to a very small value (the value of −228.59925 dBm/Hz corresponds to a resistor temperature of 0.001 Kelvin).

6. The Power parameter is used to set the modulator's output RF power. This is true for an ideal transmitted signal (no impairments added) or when small impairments are added. If large impairments are added to the signal, especially by using the GainImbalance, I_OriginOffset and Q_OriginOffset parameters, then the output RF power may be different from the value of the Power parameter.
7. Information regarding time domain signal differences between ADS Ptolemy simulations and Circuit Envelope and Transient simulations is given in the *Timed Synchronous Dataflow* (ptolemy) section in the *ADS Ptolemy Simulation* (ptolemy) documentation.

RF_ModFIR



Description: RF modulator with user defined FIR filter taps
Library: Timed, RF Subsystems
Class: TSDFRF_ModFIR
Derived From: baseSubcircuit

Parameters

Name	Description	Default	Sym	Unit	Type	Range
ROut	output resistance	DefaultROut		Ohm	real	(0, ∞)
FCarrier	carrier frequency	2e9		Hz	real	(0, ∞)
Power	modulator output power	0.01	P	W	real	[0, ∞)
VRef	reference voltage for output power calibration	1		V	real	(0, ∞)
SymbolRate	symbol rate	25e3		Hz	real	(0, ∞)
SampPerSym	samples per symbol period	8			int	[1, ∞)
FilterTaps	FIR filter taps	< $HPEESOF_DIR/adsptolemy/lib/data /square_root_raised_cosine.txt			real array	See Note 3
PhasePolarity	if set to Invert, Q channel signal is inverted: Normal, Invert	Normal			enum	
GainImbalance	gain imbalance in dB, Q channel relative to I channel	0.0			real	(-∞, ∞)
PhaseImbalance	phase imbalance in degrees, Q channel relative to I channel	0.0			real	(-∞, ∞)
I_OriginOffset	I origin offset in percent with respect to output rms voltage	0.0			real	(-∞, ∞)
Q_OriginOffset	Q origin offset in percent with respect to output rms voltage	0.0			real	(-∞, ∞)
IQ_Rotation	IQ rotation, in degress	0.0			real	(-∞, ∞)
NDensity	noise spectral density at output, in dBm/Hz	-173.975			real	(-∞, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	I_in	I input	real
2	Q_in	Q input	real

Pin Outputs

Pin	Name	Description	Signal Type
3	RF_out	output RF signal	timed

Notes/Equations

1. RF_ModFIR is a hierarchical model composed of other components. Its two inputs are baseband I and Q signals with only 1 sample per symbol. The input signals are upsampled, filtered and then used to modulate the in-phase and quadrature phase carriers of a QAM modulator. For each input sample consumed, SampPerSym output samples are produced.
2. The SymbolRate and SampPerSym parameters are used to set the simulation time step at the output of the component. The simulation time step is set to 1 / (SymbolRate × SampPerSym).
3. The FilterTaps parameter is used to set the coefficients of the FIR filter that filters the input signals. The filter coefficients must be normalized so that the sum of their squares equals 1. If this normalization is not done then the power at the output of the component will not be equal to the value of the Power parameter. The default value for FilterTaps corresponds to a square root raised-cosine filter with 65 taps, 8 samples per symbol and 0.5 excess bandwidth.
4. The VRef parameter is used to calibrate the modulator. VRef is the input voltage value that results in an instantaneous output power on a matched load equal to P. In order to get an average output power on a matched load equal to P, the input rms voltage must equal VRef. Therefore, in order to calibrate the modulator, VRef must be set to the input rms voltage.

   Note
   If you are having difficulty calibrating the output power of RF_ModFIR, push into it and follow the instructions shown on the schematic.

5. The PhasePolarity parameter can be used to invert the polarity of the Q channel signal before modulation. Depending on the configuration and number of mixers in the transmitter and receiver, the output of the demodulator may be inverted. If such a configuration is used, the Q channel signal can be correctly recovered by setting this parameter to Invert.
6. The I_OriginOffset, Q_OriginOffset, IQ_Rotation, GainImbalance, PhaseImbalance, and NDensity parameters are used to add certain impairments to the ideal transmitted signal. Impairments are added in the order described here.
   The filtered I and Q baseband input signals are applied to the I and Q inputs of a QAM modulator, which will apply the gain and phase imbalance to its quadrature phase input. The signal at the output of the QAM modulator is given by:

$$V_3(t) = A\left(V_1(t)\cos(\omega_c t) - gV_2(t)\sin\left(\omega_c t + \frac{\phi\pi}{180}\right)\right)$$

   where A is a scaling factor that depends on the Power, VRef and ROut parameters specified by the user, $V\,1(t)$ is the in-phase input, $V\,2(t)$ is the quadrature phase input, g is the gain imbalance

$$g = 10^{\frac{GainImbalance}{20}}$$

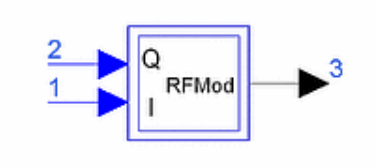
and, φ (in degrees) is the phase imbalance.
Next, the signal $V_3(t)$ is rotated by IQ_Rotation degrees. The I_OriginOffset and Q_OriginOffset are then applied to the rotated signal. Note that the amounts specified are percentages with respect to the output rms voltage. The output rms voltage is given by

$$\sqrt{2 \cdot ROut \cdot P}$$

Finally, additive noise of spectral density NDensity dBm/Hz is added to the signal. If it is necessary to generate an ideal signal I_OriginOffset, Q_OriginOffset, IQ_Rotation, GainImbalance and PhaseImbalance must all be set to zero, whereas NDensity must be set to a very small value (the value of -228.59925 dBm/Hz corresponds to a resistor temperature of 0.001 Kelvin).

7. The Power parameter is used to set the modulator output RF power. This is true for an ideal transmitted signal (no impairments added) or when small impairments are added. If large impairments are added to the signal (using GainImbalance, I_OriginOffset, and Q_OriginOffset parameters) the output RF power may be different from the value of the Power parameter.
8. Information regarding time domain signal differences between ADS Ptolemy simulations and Circuit Envelope and Transient simulations is given in the *Timed Synchronous Dataflow* (ptolemy) section in the *ADS Ptolemy Simulation* (ptolemy) documentation.

RF_Modulator



Description: RF modulator
Library: Timed, RF Subsystems
Class: TSDFRF_Modulator
Derived From: WVS

Parameters

Name	Description	Default	Unit	Type	Range
ROut	Source resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	Temperature	DefaultRTemp	Celsius	real	[-273.15, ∞)
FCarrier	Carrier frequency	1900 MHz	Hz	real	(0, ∞)
Power	Power	0.01 W	W	real	[0, ∞)
VRef	Reference voltage for output power calibration	1 V	V	real	(0, ∞)
SamplingRate	Sampling rate	25e3	Hz	real	(0, ∞)
MirrorSpectrum	Mirror spectrum about carrier? NO, YES	NO		enum	
GainImbalance	Gain imbalance in dB, Q channel relative to I channel	0.0	dB	real	(-∞, ∞)
PhaseImbalance	Phase imbalance in degrees, Q channel relative to I channel	0.0	deg	real	(-∞, ∞)
I_OriginOffset	I origin offset in percent with repect to output rms voltage	0.0		real	(-∞, ∞)
Q_OriginOffset	Q origin offset in percent with repect to output rms voltage	0.0		real	(-∞, ∞)
IQ_Rotation	IQ rotation, in degrees	0.0	deg	real	(-∞, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	I_in	I input	real
2	Q_in	Q input	real

Pin Outputs

Pin	Name	Description	Signal Type
3	out	output RF signal	timed

Notes/Equations

1. RF_Modulator is a hierarchical model composed of other components. Its two inputs are baseband I and Q signals. The input signals are assumed to be filtered with multiple samples per symbol. RF_Modulator does not upsample or filter the input signals. The input signals are used to modulate the in-phase and quadrature phase

carriers of a QAM modulator.

2. The SamplingRate parameter is used to set the simulation time step at the output of the component. The simulation time step is set to 1 / SamplingRate.
3. The VRef parameter is used to calibrate the modulator. VRef is the input voltage value that results in an instantaneous output power on a matched load equal to P. In order to get an average output power on a matched load equal to P, the input rms voltage must equal VRef. Therefore, in order to calibrate the modulator, VRef must be set to the input rms voltage. If the input rms voltage is not known, the TkIQrms component can be connected to the inputs of the RF_Modulator and it will report the rms value of the input IQ signal.
4. The MirrorSpectrum parameter can be used to mirror the spectrum (invert the Q envelope) at the output of the modulator. Depending on the configuration of the mixers in the upconverter, which typically follows a modulator, the signal at the upconverter's input may need to be mirrored. If such a configuration is used, then this parameter should be set to YES.
5. The I_OriginOffset, Q_OriginOffset, IQ_Rotation, GainImbalance, PhaseImbalance, Temp and Noise parameters are used to add certain impairments to the ideal transmitted signal. Impairments are added in the order described here.
   The I and Q baseband input signals are applied to the I and Q inputs of a QAM modulator, which applies the gain and phase imbalance to its quadrature phase input. The signal at the output of the QAM modulator is given by:

   $$V_3(t) = A\left(V_1(t)\cos(\omega_c t) - gV_2(t)\sin\left(\omega_c t + \frac{\phi\pi}{180}\right)\right)$$

   where A is a scaling factor that depends on the Power, VRef and R parameters specified by the user, $V_1(t)$ is the in-phase input, $V_2(t)$ is the quadrature phase input, g is the gain imbalance

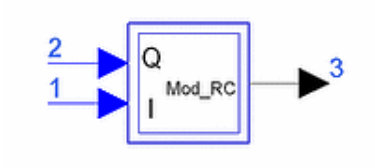
   $$g = 10^{\frac{GainImbalance}{20}}$$

   and φ (in degrees) is the phase imbalance.
   Next, the signal $V_3(t)$ is rotated by IQ_Rotation degrees. The I_OriginOffset and Q_OriginOffset are then applied to the rotated signal. Note that the amounts specified are percentages with respect to the output rms voltage. The output rms voltage is given by

   $$\sqrt{2 \cdot ROut \cdot P}$$

   Finally, additive white gaussian noise is added to the signal. The noise added is thermal noise generated by the output resistor (R parameter) based on the resistor temperature (Temp parameter). Noise is added only when the Noise parameter is set to YES and the Temp parameter is greater than -273.15 ^{o} C.
   If it is necessary to generate an ideal signal I_OriginOffset, Q_OriginOffset, IQ_Rotation, GainImbalance, and PhaseImbalance must all be set to zero, and Noise must be set to NO.
6. The Power parameter is used to set the modulator output RF power. This is true for an ideal transmitted signal (no impairments added) or when small impairments are added. If large impairments are added to the signal (using GainImbalance, I_OriginOffset, and Q_OriginOffset parameters) the output RF power may be different from the value of the Power parameter.
7. Information regarding time domain signal differences between ADS Ptolemy simulations and Circuit Envelope and Transient simulations is given in the *Timed Synchronous Dataflow* (ptolemy) section in the *ADS Ptolemy Simulation* (ptolemy) documentation.

RF_ModulatorRC



Description: RF modulator with raised cosine filters
Library: Timed, RF Subsystems
Class: TSDFRF_ModulatorRC
Derived From: RF_Modulator

Parameters

Name	Description	Default	Unit	Type	Range
ROut	Source resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	Temperature	DefaultRTemp	Celsius	real	[-273.15, ∞)
FCarrier	Carrier frequency	1900 MHz	Hz	real	(0, ∞)
Power	Power	0.01 W	W	real	[0, ∞)
VRef	Reference voltage for output power calibration	1 V	V	real	(0, ∞)
SymbolRate	Symbol rate	25e3	Hz	real	(0, ∞)
SampPerSym	Samples per symbol period	8		int	[1, ∞)
FilterLength	Length of raised cosine filters in number of symbols	16		int	(0, ∞)
ExcessBW	Raised cosine filter excess bandwidth	0.35		real	[0, 1]
SquareRoot	Square-root raised-cosine model? NO, YES	YES		enum	
MirrorSpectrum	Mirror spectrum about carrier? NO, YES	NO		enum	
GainImbalance	Gain imbalance in dB, Q channel relative to I channel	0.0	dB	real	(-∞, ∞)
PhaseImbalance	Phase imbalance in degrees, Q channel relative to I channel	0.0	deg	real	(-∞, ∞)
I_OriginOffset	I origin offset in percent with repect to output rms voltage	0.0		real	(-∞, ∞)
Q_OriginOffset	Q origin offset in percent with repect to output rms voltage	0.0		real	(-∞, ∞)
IQ_Rotation	IQ rotation, in degrees	0.0	deg	real	(-∞, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	I_in	I input	real
2	Q_in	Q input	real

Pin Outputs

Pin	Name	Description	Signal Type
3	out	output RF signal	timed

Notes/Equations

1. RF_ModulatorRC is a hierarchical model composed of other components. Its two inputs are baseband I and Q signals with only one sample per symbol. The input signals are upsampled, filtered and then used to modulate the in-phase and quadrature phase carriers of a QAM modulator. For each input sample consumed, SampPerSym output samples are produced.
2. The SymbolRate and SampPerSym parameters are used to set the simulation time step at the output of the component. The simulation time step is set to 1 / (SymbolRate × SampPerSym).
3. The filter used to filter the input I and Q signals is a raised cosine one. The filter length, excess bandwidth, and type (raised cosine or root raised cosine) can be specified using the parameters FilterLength, ExcessBW, and SquareRoot respectively.
4. The VRef parameter is used to calibrate the modulator. VRef is the input voltage value that results in an instantaneous output power on a matched load equal to P. In order to get an average output power on a matched load equal to P, the input rms voltage must equal VRef. Therefore, in order to calibrate the modulator, VRef must be set to the input rms voltage. If the input rms voltage is not known, the TkIQrms component can be connected to the inputs of the RF_Modulator and it will report the rms value of the input IQ signal.
5. The MirrorSpectrum parameter can be used to mirror the spectrum (invert the Q envelope) at the output of the modulator. Depending on the configuration of the mixers in the upconverter, which typically follows a modulator, the signal at the upconverter's input may need to be mirrored. If such a configuration is used, then this parameter should be set to YES.
6. The I_OriginOffset, Q_OriginOffset, IQ_Rotation, GainImbalance, PhaseImbalance, Temp and Noise parameters are used to add certain impairments to the ideal transmitted signal. Impairments are added in the order described here.
   The filtered I and Q baseband input signals are applied to the I and Q inputs of a QAM modulator, which applies the gain and phase imbalance to its quadrature phase input. The signal at the output of the QAM modulator is given by:

   $$V_3(t) = A\left(V_1(t)\cos(\omega_c t) - gV_2(t)\sin\left(\omega_c t + \frac{\phi\pi}{180}\right)\right)$$

   where A is a scaling factor that depends on the Power, VRef and R parameters specified by the user, $V_1(t)$ is the in-phase input, $V_2(t)$ is the quadrature phase input, g is the gain imbalance

   $$g = 10^{\frac{GainImbalance}{20}}$$

   and φ (in degrees) is the phase imbalance.
   Next, the signal $V_3(t)$ is rotated by IQ_Rotation degrees. The I_OriginOffset and Q_OriginOffset are then applied to the rotated signal. Note that the amounts specified are percentages with respect to the output rms voltage. The output rms voltage is given by

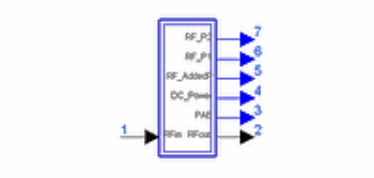
   $$\sqrt{2 \times ROut \times P}$$

   Finally, additive white gaussian noise is added to the signal. The noise added is thermal noise generated by the output resistor (R parameter) based on the resistor temperature (Temp parameter). Noise is added only when the Noise parameter is set to YES and the Temp parameter is greater than -273.15 ° C.
   If it is necessary to generate an ideal signal I_OriginOffset, Q_OriginOffset, IQ_Rotation, GainImbalance, and PhaseImbalance must all be set to zero, and Noise

must be set to NO.

7. The Power parameter is used to set the modulator output RF power. This is true for an ideal transmitted signal (no impairments added) or when small impairments are added. If large impairments are added to the signal (using GainImbalance, I_OriginOffset, and Q_OriginOffset parameters) the output RF power may be different from the value of the Power parameter.
8. Information regarding time domain signal differences between ADS Ptolemy simulations and Circuit Envelope and Transient simulations is given in the *Timed Synchronous Dataflow* (ptolemy) section in the *ADS Ptolemy Simulation* (ptolemy) documentation.

RF_PAE_TestFixture



Description: RF power amplifier test fixture to measure power added efficiency
Library: Timed, RF Subsystems
Class: TSDFRF_PAE_TestFixture

Parameters

Name	Description	Default	Unit	Type	Range
TStep	system level simulation time step	0.001	sec	real	(0, ∞)
FCarrier	input signal RF carrier frequency	1000000	Hz	real	(0, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	RFin	RF input	timed

Pin Outputs

Pin	Name	Description	Signal Type
2	RFout	RF output	timed
3	PAE	power added efficiency for PA tested	real
4	DC_Power	dc power (Watts) consumed by PA tested	real
5	RF_AddedPower	RF power added (Watts) by PA tested	real
6	RF_P1	RF power (Watts) for input RF signal	real
7	RF_P2	RF power (Watts) for output RF signal	real

Notes/Equations

1. RF_PAE_TestFixture is a hierarchical model composed of other components. Inside this test fixture is an RF power amplifier design from which up to 6 DC bias sense voltages and currents are made available for use in power-added efficiency measurements.
2. This test fixture design contains all EnvOutSelector components required at all outputs of the Analog/RF Schematic design to be tested.
3. Push into RF_PAE_TestFixture to see the CktRF_PA circuit design for which PAE is to be measured.
   Replace CktRF_PA with your specific RF circuit power amplifier design to measure its PAE.
   The RF circuit power amplifier design must route to its internal ports (thus, its external symbol pins) the sense values for the circuit V_DC source voltage and load current values.
   If the user-defined power amplifier design has an internal number of V_DC sources other than 4, the user should save this RF_PAE_TestFixture to their local workspace and modify the design to work with the user-defined number of V_DC sense voltage

and current pairs.

4. The PAE is calculated as the ratio of RF power added by the PA divided by DC power consumed by the PA.
5. DC power consumed by the PA is the sum of the DC bias powers consumed by the V_DC sources within the CktRF_PA design.
6. The RF_PAE_TestFixture outputs values for PAE, DC_Power, RF_AddedPower, RF power at amplifier input (RF_P1), and RF power at amplifier output (RF_P2).
7. Use of this component is demonstrated in the *File* > *Open* > *Example* > PtolemyDocExamples > Timed_RF_Subsystems_wrk_. Open the networks design *RF_PAE_example* where RF_PAE_TestFixture is used.
8. Information regarding time domain signal differences between ADS Ptolemy simulations and Circuit Envelope and Transient simulations is given in the *Timed Synchronous Dataflow* (ptolemy) section in the *ADS Ptolemy Simulation* (ptolemy) documentation.

RF_RX_IFout

Description: Double down-conversion RF receiver with IF output
Library: Timed, RF Subsystems
Class: TSDFRF_RX_IFout
Derived From: baseSubcircuit

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RX_AntTemp	receiving antenna noise temperature, in Kelvin	150.0		real	[0, ∞)
RX_Gain	receiver power gain, in dB	50.0		real	(-∞, ∞)
RX_NF	receiver noise figure, in dB	5.0		real	[0, ∞)
RF_Freq	RF frequency	900e6	Hz	real	(0, ∞)
RF_BW	RF filter bandwidth	25e6	Hz	real	(0, ∞)
IF_Freq1	first IF frequency	100e6	Hz	real	(0, ∞)
IF_Freq2	second IF frequency	400e3	Hz	real	(0, ∞)
IF_BW	IF bandwidth	30e3	Hz	real	(0, ∞)
IP3in	IP3 at receiver input	3.2e-6	W	real	(-∞, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	RF	input signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
2	IF	output signal	timed

Notes/Equations

1. RF_RX_IFout is a hierarchical model composed of other components. The receiver is used to convert input RF signal to output IF signal with nonlinear distortion and additive noise.
2. RX_Gain and IP3in parameters determine the nonlinear distortion. RX_AntTemp and RX_NF parameters determine the additive noise.
3. This component uses the double down-conversion (super-heterodyne) scheme. Low-side LO signals are used. Consequently there is no spectral inversion at the output.
4. If RF_BW is much greater than IF_BW, and the simulation time step is set according to IF_BW, a warning message from the RF filter may be issued. This is because RF filter bandwidth is not fully characterized. This usually does not affect the simulation

accuracy.

5. The output signal-noise ratio is determined as follows:
   S/N (dB)=S-N
   S (dBm) = P_in (dBm) + RX_Gain (dB)
   N (dBm) = N0_out (dBm) + 10 × log10 (IF_BW (Hz))
   N0_out (dBm) = N0_in (dBm) + RX_Gain (dB) +RX_NF (dB)
   N0_in (dBm) = 10 × log10 (K × RX_AntTemp) + 30dB
   K = Boltzmann's constant
6. Information regarding time domain signal differences between ADS Ptolemy simulations and Circuit Envelope and Transient simulations is given in the *Timed Synchronous Dataflow* (ptolemy) section in the *ADS Ptolemy Simulation* (ptolemy) documentation.

RF_RX_IFout1

Description: Single down-conversion RF receiver with IF output
Library: Timed, RF Subsystems
Class: TSDFRF_RX_IFout1
Derived From: baseSubcircuit

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RX_AntTemp	receiving antenna noise temperature, in Kelvin	150.0		real	[0, ∞)
RX_Gain	receiver power gain, in dB	50.0		real	(-∞, ∞)
RX_NF	receiver noise figure, in dB	5.0		real	[0, ∞)
RF_Freq	RF frequency	900e6	Hz	real	(0, ∞)
RF_BW	RF filter bandwidth	25e6	Hz	real	(0, ∞)
IF_Freq	IF frequency	400e3	Hz	real	(0, ∞)
IF_BW	IF bandwidth	30e3	Hz	real	(0, ∞)
IP3in	IP3 at receiver input	3.2e-6	W	real	(-∞, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	RF	input signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
2	IF	output signal	timed

Notes/Equations

1. RF_RX_IFout1 is a hierarchical model composed of other components. The receiver is used to convert an input RF signal to an output IF signal with nonlinear distortion and additive noise.
2. RX_Gain and IP3in parameters determine the nonlinear distortion. RX_AntTemp and RX_NF parameters determine the additive noise.
3. This component uses the single down-conversion scheme. A low-side LO signal is used. Therefore, there is no spectral inversion at the output.
4. If RF_BW is much greater than IF_BW, and the simulation time step is set according to IF_BW, a warning message from the RF filter may be issued. This is because RF filter bandwidth is not fully characterized. This usually does not affect the simulation accuracy.
   The output signal-noise ratio is determined as follows:

5.
   S/N (dB)=S-N
   S (dBm) = P_in (dBm) + RX_Gain (dB)
   N (dBm) = N0_out (dBm) + 10 × log10 (IF_BW (Hz))
   N0_out (dBm) = N0_in (dBm) + RX_Gain (dB) +RX_NF (dB)
   N0_in (dBm) = 10 × log10 (K × RX_AntTemp) + 30dB
   K = Boltzmann's constant
6. Information regarding time domain signal differences between ADS Ptolemy simulations and Circuit Envelope and Transient simulations is given in the *Timed Synchronous Dataflow* (ptolemy) section in the *ADS Ptolemy Simulation* (ptolemy) documentation.

RF_RX_IFout_SBlock

Description: RF receiver with IF output and user specified S21 parameter file for the IF filter
Library: Timed, RF Subsystems
Class: TSDFRF_RX_IFout_SBlock
Derived From: baseSubcircuit

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RX_AntTemp	receiving antenna noise temperature, in Kelvin	150.0		real	[0, ∞)
RX_Gain	receiver power gain, in dB	50.0		real	(-∞, ∞)
RX_NF	receiver noise figure, in dB	5.0		real	[0, ∞)
RF_Freq	RF frequency	900e6	Hz	real	(0, ∞)
RF_BW	RF filter bandwidth	25e6	Hz	real	(0, ∞)
IF_Freq	IF frequency	100e6	Hz	real	(0, ∞)
File	file for S21 data in Touchtone format	myfile.s2p		string	
N	number of points for S21 data impulse response	2048		int	[2, ∞)
IP3in	IP3 at receiver input	3.2e-6	W	real	(-∞, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	RF	input signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
2	IF	output signal	timed

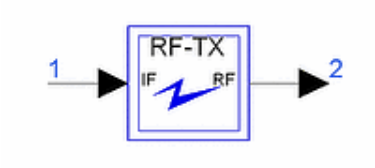
Notes/Equations

1. RF_RX_IFout_SBlock is a hierarchical model composed of other components. The receiver is used to convert input RF signal to output IF signal with nonlinear distortion and additive noise.
2. RX_Gain and IP3in parameters determine the nonlinear distortion. RX_AntTemp and RX_NF parameters determine the additive noise.
3. This component uses the single down-conversion scheme. Low-side LO signal is used. Consequently there is no spectral inversion at the output.
4. Typically, RF_BW should be set greater than the bandwidth defined by the IF S-parameter-based filter BW, and the simulation time step is set according to IF

bandwidth, a warning message from the RF filter may be issued. This is because RF filter bandwidth is not fully characterized. This usually does not affect the simulation accuracy.

5. The output signal-noise ratio is determined as follows:
   S/N (dB)= S-N
   S (dBm) = P_in (dBm) + RX_Gain (dB)
   N (dBm) = N0_out (dBm) + 10 × log10 (IF_BW (Hz))
   I_BW(Hz) = noise bandwidth of the S-parameter-based filter
   N0_out (dBm) = N0_in (dBm) + RX_Gain (dB) +RX_NF (dB)
   N0_in (dBm) = 10 × log10 (K × RX_AntTemp) + 30dB
   K = Boltzmann's constant
6. The IF filter is modeled by the S-parameters for a bandpass filter that the user specifies by entering the filename for S-parameter data. The data file typically has a *.s2p* filename extension. For details on this file format, refer to the *S2PMDIF Format* (cktsim) in the *Using Circuit Simulators* (cktsim) documentation.
7. Information regarding time domain signal differences between ADS Ptolemy simulations and Circuit Envelope and Transient simulations is given in the *Timed Synchronous Dataflow* (ptolemy) section in the *ADS Ptolemy Simulation* (ptolemy) documentation.

RF_TX_IFin



Description: RF transmitter with IF input and 2 filter-amplifier pairs
Library: Timed, RF Subsystems
Class: TSDFRF_TX_IFin
Derived From: baseSubcircuit

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
IF_Freq	IF frequency	400e3	Hz	real	(0, ∞)
RF_Freq	RF frequency	900e6	Hz	real	(0, ∞)
RF_BW	RF bandwidth	30e3	Hz	real	(0, ∞)
TX_Gain	transmitter power gain, in dB	80		real	(-∞, ∞)
PSat	saturated output power	32	W	real	(0, ∞)
NDensity	noise spectral density at output, in dBm/Hz	-173.975		real	(-∞, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	IF	input signal	timed

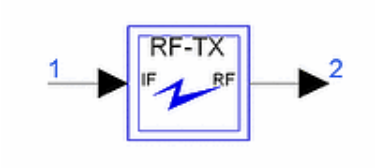
Pin Outputs

Pin	Name	Description	Signal Type
2	RF	output signal	timed

Notes/Equations

1. RF_TX_IFin is a hierarchical model composed of other components. The transmitter is used to convert input IF signal to output RF signal with nonlinear distortion and additive noise.
2. Nonlinear distortion is determined by the PSat parameter. PSat models am-am distortion only (it does not model am-pm distortion). For modeling am-pm distortion, use of the GComp parameter on the internal GainRF component with instance name TxPowerAmp is recommended.
3. Noise is modeled by the NDensity parameter. No other component in the subnetwork produces any additive noise.
4. Information regarding time domain signal differences between ADS Ptolemy simulations and Circuit Envelope and Transient simulations is given in the *Timed Synchronous Dataflow* (ptolemy) section in the *ADS Ptolemy Simulation* (ptolemy) documentation.

RF_TX_IFin1



Description: RF transmitter with IF input and 1 filter-amplifier pair
Library: Timed, RF Subsystems
Class: TSDFRF_TX_IFin1
Derived From: baseSubcircuit

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
IF_Freq	IF frequency	400e3	Hz	real	(0, ∞)
RF_Freq	RF frequency	900e6	Hz	real	(0, ∞)
RF_BW	RF bandwidth	30e3	Hz	real	(0, ∞)
TX_Gain	transmitter power gain, in dB	80		real	(-∞, ∞)
PSat	saturated output power	32	W	real	(0, ∞)
NDensity	noise spectral density at output, in dBm/Hz	-173.975		real	(-∞, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	IF	input signal	timed

Pin Outputs

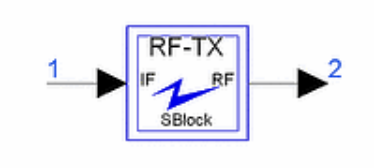
Pin	Name	Description	Signal Type
2	RF	output signal	timed

Notes/Equations

1. RF_TX_IFin1 is a hierarchical model composed of other components. The transmitter is used to convert an input IF signal to an output RF signal with nonlinear distortion and additive noise.
2. Nonlinear distortion is determined by the PSat parameter. PSat models am-am distortion only (it does not model am-pm distortion). For modeling am-pm distortion, use of the GComp parameter on the internal GainRF component with instance name TxPowerAmp is recommended.
3. Noise is modeled by the NDensity parameter. No other component in the subnetwork produces any additive noise.
4. The only difference between RF_TX_IFin1 and RF_TX_IFin is that RF_TX_IFin1 uses one filter and one amplifier, whereas RF_TX_IFin uses two filters and two amplifiers.
5. Information regarding time domain signal differences between ADS Ptolemy simulations and Circuit Envelope and Transient simulations is given in the *Timed Synchronous Dataflow* (ptolemy) section in the *ADS Ptolemy Simulation* (ptolemy)

documentation.

RF_TX_IFin_SBlock



Description: RF transmitter with IF input and user specified S21 parameter file for TX filter
Library: Timed, RF Subsystems
Class: TSDFRF_TX_IFin_SBlock
Derived From: baseSubcircuit

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
IF_Freq	IF frequency	400e3	Hz	real	(0, ∞)
RF_Freq	RF frequency	900e6	Hz	real	(0, ∞)
File	file for S21 data in Touchtone format	myfile.s2p		string	
N	number of points for S21 data impulse response	2048		int	[2, ∞)
TX_Gain	transmitter power gain, in dB	80		real	(-∞, ∞)
PSat	saturated output power	32	W	real	(0, ∞)
NDensity	noise spectral density at output, in dBm/Hz	-173.975		real	(-∞, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	IF	input signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
2	RF	output signal	timed

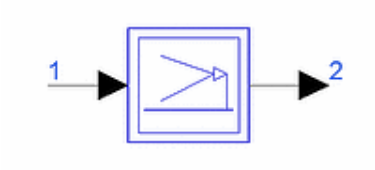
Notes/Equations

1. RF_TX_IFin_SBlock is a hierarchical model composed of other components. The transmitter is used to convert input IF signal to output RF signal with nonlinear distortion and additive noise.
2. Nonlinear distortion is determined by the PSat parameter. PSat models am-am distortion only (it does not model am-pm distortion). For modeling am-pm distortion, use of the GComp parameter on the internal GainRF component with instance name TxPowerAmp.
3. Noise is modeled by the NDensity parameter. No other component in the subnetwork produces any additive noise.
4. The IF filter is modeled by the S-parameters for a bandpass filter that the user specifies by entering the filename for S-parameter data. The data file typically has an *.s2p* filename extension. For details on this file format, refer to *S2PMDIF Format*

(cktsim) in the *Using Circuit Simulators* (cktsim) documentation.

5. Information regarding time domain signal differences between ADS Ptolemy simulations and Circuit Envelope and Transient simulations is given in the *Timed Synchronous Dataflow* (ptolemy) section in the *ADS Ptolemy Simulation* (ptolemy) documentation.

RxAntTempK



Description: Add noise to input signal due to the receiver antenna temperature
Library: Timed, RF Subsystems
Class: TSDF_RxAntTempK
Derived From: baseSubcircuit

Parameters

Name	Description	Default	Unit	Type	Range
RIn	input resistance	DefaultRIn	Ohm	real	(0, ∞)
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
TempK	receiving antenna noise temperature, in Kelvin	290		real	[0, ∞)

Pin Inputs

Pin	Name	Description	Signal Type
1	input	input signal	timed

Pin Outputs

Pin	Name	Description	Signal Type
2	output	output signal	timed

Notes/Equations

1. RxAntTempK models the receiver antenna noise temperature by adding white Gaussian noise to the input signal. The input signal power is preserved when the component is connected to a matched load. The noise power delivered to a matched load is k × TempK × BW, where k=1.380622e-23 is Boltzmann's constant and BW is the simulation bandwidth. (BW=1/TStep, where TStep is the simulation time step of the input signal.)
   The noise contributed is an independent noise process. This noise is dependent on the value of the DefaultSeed in the Data Flow controller. When DefaultSeed=0, then the noise generated for each simulation is different; when DefaultSeed>0, then the noise generated for each simulation, though random, has the same initial seed starting condition and thus results in reproducible noise for each simulation.
2. Information regarding time domain signal differences between ADS Ptolemy simulations and Circuit Envelope and Transient simulations is given in the *Timed Synchronous Dataflow* (ptolemy) section in the *ADS Ptolemy Simulation* (ptolemy) documentation.

Timed Sources

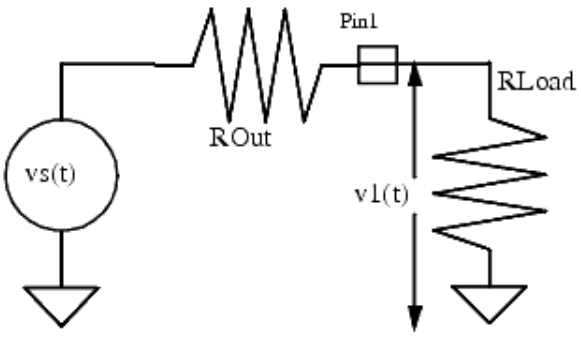
The Timed Sources library contains baseband and RF (complex envelope) timed signal sources. All sources have a specified time step (TStep) that sets the time increment for each firing of the model during simulation.

- When TStep is set to 0 it assumes the time step value associated with any other timed signal that propagates throughout the design network.
- When TStep is set to 0 and no other timed signal exists in the network, an error will be declared and the simulation will terminate.

All sources (except *ConstTimed* (timed)) have a user-specified output resistance (ROut) for a series resistor at the component output pin.

When ROut>DF_ZERO_OHMS (defined as 1.0e-12), the output signal source vs(t) has a value equal to twice the output pin signal v1(t) when the output is connected to a matched resistor load. The circuit model below demonstrates this.

Timed Sources Circuit Model



$v1(t) = vs(t) * RLoad / (RLoad + ROut)$

$v1(t) = vs(t)/2$ when $RLoad = ROut$

The output pin signal v1(t) at the output series resistance is dependent on the value of the load resistance connected to it. When load resistor RLoad is equal to the model ROut the value of v1(t) is equal to vs(t)/2, otherwise, based on the voltage divider action, v1(t) is:

$V1(t) = Vs(t) \times RLoad/(RLoad + ROut)$

where *V1* and *Vs* are related to their complex envelopes v1(t) and vs(t) at their mathematical characterization frequency fc, as follows:

$$V_1(t) = Re\left\{(v_1(t))e^{j2\pi f_c t}\right\} = Re\left\{(v_{1I}(t) + jv_{1Q}(t))e^{j2\pi f_c t}\right\}$$

$$V_s(t) = Re\left\{(v_s(t))e^{j2\pi f_c t}\right\} = Re\left\{(v_{sI}(t) + jv_{sQ}(t))e^{j2\pi f_c t}\right\}$$

This mathematical representation is valid if the simulation time step is less than the inverse of the characterization frequency and the signal information content has an information bandwidth less than the characterization frequency. Signal source signals in this section are defined by their expression for vs(t).

When ROut≤DF_ZERO_OHMS, the above discussions are valid except the value of vs(t) will be equal to one-half of the vs(t) value of the case ROut > DF_ZERO_OHMS. This is not a common case and must be avoided unless absolutely necessary.

The output resistor contributes additive thermal noise (kTB) to the output signal when the specified resistance temperature (RTemp) is greater than absolute zero (−273.15 °C) where:

k = Boltzmann's constant
T = temperature in Kelvin
B = simulation frequency bandwidth:

- 1/2/tstep if signal is a timed baseband signal;
- 1/tstep if signal is a timed complex envelope signal

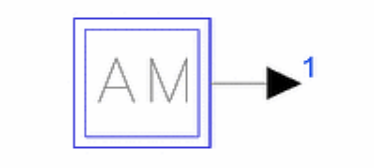
When RTemp > −273.15, noise contributed from each resistor instance is an independent noise process. This noise is dependent on the value of the DefaultSeed in the DF controller. For more information, refer to *DF (Data Flow) Controller* (ptolemy) in the *ADS Ptolemy Simulation* (ptolemy) documentation. When DefaultSeed=0, noise generated for each simulation is different; when DefaultSeed > 0, noise generated for each simulation, though random, has the same initial seed starting condition and results in reproducible noise for each simulation.

Note
Information regarding time domain signal differences between ADS Ptolemy simulations and Circuit Envelope and Transient simulations is given in the *Timed Synchronous Dataflow* (ptolemy) section in the *ADS Ptolemy Simulation* (ptolemy) documentation.

Components

- *AM* (timed)
- *Clock* (timed)
- *ConstTimed* (timed)
- *Data* (timed)
- *FM* (timed)
- *Impulse* (timed)
- *Noise* (timed)
- *NoiseFMask* (timed)
- *N Tones* (timed)
- *PM* (timed)
- *Pulse* (timed)
- *PulseRF* (timed)
- *QAM* (timed)
- *Ramp* (timed)
- *Sinusoid* (timed)
- *TimedDataRead* (timed)
- *TimedExpression* (timed)
- *TimedSource* (timed)
- *Video* (timed)

AM



Description: Amplitude modulated carrier with single modulating tone
Library: Timed, Sources
Class: TSDF_AM
Derived From: basesource
C++ Code: See *doc/sp_items/TSDF_AM.html* under your installation directory.

Parameters

Name	Description	Default	Unit	Type	Range
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
TStep	simulation time step	0.0001	sec	real	[0, ∞)
Type	type of AM: Conventional Am, DSB SC Am	Conventional Am		enum	
FCarrier	carrier frequency	1000000.0	Hz	real	(0, ∞)
Power	unmodulated carrier power	.010	W	real	[0, ∞)
Phase	unmodulated carrier phase, in degrees	0.0	deg	real	(-∞, ∞)
VRef	modulator reference voltage level	1.0	V	real	(0, ∞)
FSignal	modulation signal frequency	1000	Hz	real	1/(2 * TStep), ∞)
VPeak	peak modulation voltage level	1.0	V	real	(-∞, ∞)

Pin Outputs

Pin	Name	Description	Signal Type
1	output	output signal	timed

Notes/Equations

1. For information regarding timed source signals, refer to *Timed Sources* (timed).
2. This source generates an RF (complex envelope) timed signal output at a characterization frequency of *FCarrier*. It is an amplitude modulated carrier whose modulating signal is a single tone.
   For Type = Conventional Am:
   $$v = 1 + \frac{VPeak}{VRef} \times \cos(2 \times \pi \times FSignal \times t)$$
   For Type = DSB-SC Am:
   $$v = \frac{VPeak}{VRef} \times \cos(2 \times \pi \times FSignal \times t)$$
   The output signal is given by
   $$vs = 2 \times \sqrt{2 \times Power \times ROut} \times v \times e^{j \times \pi \times \frac{Phase}{180}}$$

3. Note that the Power parameter sets the unmodulated carrier power. The actual modulator output power is given by

   $$Power \times V^2_{rms}$$

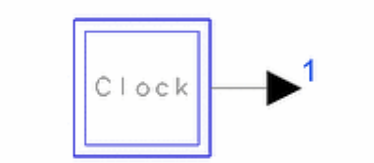
   For Type = Conventional Am:

   $$V_{rms} = \sqrt{1 + VPeak^2 / (VRef^2 \times 2)}$$

   For Type = DSB-SC Am:

   $$V_{rms} = VPeak / (VRef \times \sqrt{2})$$

Clock



Description: Clock generator
Library: Timed, Sources
Class: TSDF_Clock
Derived From: basesource
C++ Code: See *doc/sp_items/TSDF_Clock.html* under your installation directory.

Parameters

Name	Description	Default	Unit	Type	Range
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
TStep	simulation time step	0.0001	sec	real	[0, ∞)
Period	clock time period	.0006	sec	real	(0, ∞)†
Delay	time delay before turn on	.0002	sec	real	[0, ∞)
DutyCycle	clock duty cycle	0.5		real	(0:1)†

† This parameter must satisfy the condition TStep ≤ min(DutyCycle × Period, (1-DutyCycle) × Period)

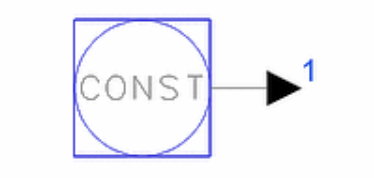
Pin Outputs

Pin	Name	Description	Signal Type
1	output	output signal	timed

Notes/Equations

1. For information regarding timed source signals, refer to *Timed Sources* (timed).
2. This source generates a baseband timed signal output. It is a clock signal with duty cycle that can be set by the user.
   Let normalized time = ntime = fmod(t−Delay, Period)
   if ((ntime<0.0) or (ntime+TStep/2> Period))
   ntime = 0
   if ((t + TStep/2 >= Delay) and (ntime + TStep/2 < DutyCycle × Period))
   v = 1
   else
   v = 0
   if ROut = 0; then vs = v
   if ROut > 0; then vs = 2×v

ConstTimed



Description: Constant timed signal generator
Library: Timed, Sources
Class: TSDFConstTimed
Derived From: basesource
C++ Code: See *doc/sp_items/TSDFConstTimed.html* under your installation directory.

Parameters

Name	Description	Default	Unit	Type	Range
TStep	simulation time step	0.0001	sec	real	[0, ∞)
FCarrier	carrier frequency	0.0	Hz	real	[0, ∞)
Value	signal value	1.0		complex	

Pin Outputs

Pin	Name	Description	Signal Type
1	output	output signal	timed

Notes/Equations

1. For information regarding timed source signals, refer to *Timed Sources* (timed).
2. This source generates a constant baseband signal when FCarrier=0, or a constant RF signal when FCarrier > 0.
   When FCarrier = 0, then vs = Real (value)
   When FCarrier > 0, then vs = Value
3. For details on complex parameter values, refer to *Complex-Valued Parameters* (ptolemy) in the *ADS Ptolemy Simulation* (ptolemy) documentation.

Data

Description: Data generator
Library: Timed, Sources
Class: TSDF_Data
Derived From: basesource

Parameters

Name	Description	Default	Unit	Type	Range
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
TStep	simulation time step	0.0001	sec	real	[0, ∞)
BitTime	Time interval for one bit	0.001	sec	real	[TStep, ∞)
UserPattern	user specified bit pattern in binary (prefix %), octal (prefix 0), or hex (prefix 0x)			string array	
Type	Data sequence type random or prbs: Random, Prbs	Prbs		enum	
SequencePattern	sequence pattern	8		int	[2:23]
Repeat	Yes or No to define sequence as time periodic signal: No, Yes	Yes		enum	

Pin Outputs

Pin	Name	Description	Signal Type
1	output	output signal	timed

Notes/Equations

Note
Due to a corrective change made to this component in the ADS 2008 release, its output bit sequence in ADS 2008 (and later releases) differs from the one generated in previous releases when Type=Prbs and SequencePattern is 5, 8, 9, 10, 11, 12, 13, or 14. In releases prior to ADS 2008, the bit sequences generated with these settings did not have the proper statistics (mean and autocorrelation). This problem was fixed in ADS 2008.

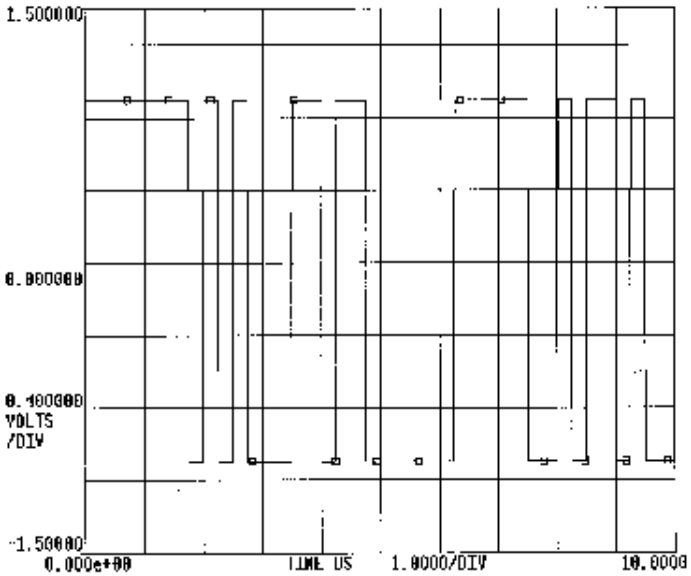
1. For information regarding timed source signals, refer to *Timed Sources* (timed).
2. The Data source generates a user-defined or random baseband NRZ waveform.
3. If UserPattern is set, the output of the source will be user-defined; in this case, the Type and SequencePattern parameters are ignored.
   The UserPattern parameter is an array of strings defining a bit pattern. Each string can be given in hexadecimal, octal, or binary format. To enter a string in hexadecimal, octal or binary format prefix it with 0x, 0, or % respectively; the % sign can be omitted if the binary string starts with 1. As for all array type

parameters, entry from a file is also supported, for example UserPattern="<pattern.txt".
The NRZ waveform is generated by first converting all strings in binary and then mapping 0s to -1s and 1s to 1s. For example, if UserPattern="0472 %0010 0xA3F 1101", the corresponding bit pattern is 1 0 0 1 1 1 0 1 0 0 0 1 0 1 0 1 0 0 0 1 1 1 1 1 1 1 1 0 1 and the generated NRZ waveform will be 1 -1 -1 1 1 1 -1 1 -1 -1 -1 1 -1 1 -1 1 -1 -1 -1 1 1 1 1 1 1 1 1 -1 1.
If UserPattern is empty then the output of the source is random and its properties are defined by the Type and SequencePattern parameters. SequencePattern sets the period to $2^{SequencePattern}$.

- If Type is set to Prbs, the output of each Data instance with the same value for SequencePattern is the same pseudorandom sequence.
- If Type is set to Random, the output from each Data instance is independent. In this case, the output sequences depend on the value of DefaultSeed in the DF (data flow) controller. When DefaultSeed=0, the random sequences will differ from one simulation to another; when DefaultSeed > 0, the random sequences will be the same for every simulation.

4. The Repeat parameter is used to turn on or off the periodicity of the source's output. If Repeat is set to Yes, the signal will be periodic with period $2^{SequencePattern}$ or the length of the UserPattern, otherwise the output will be 0 after one repetition.
5. A data generator binary signal is illustrated below.

Data Generator Binary Signal; BitTime=0.25 µsec, TStep=0.025 µsec



6. As an alternate to this Data source, a Pulse or Clock source can be used with an LFSR (linear feedback shift register) component to generate a pseudo random data stream with user-specified feedback taps.

FM

Description: Frequency modulated carrier with single modulating tone
Library: Timed, Sources
Class: TSDF_FM
Derived From: basesource
C++ Code: See *doc/sp_items/TSDF_FM.html* under your installation directory.

Parameters

Name	Description	Default	Unit	Type	Range
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
TStep	simulation time step	0.0001	sec	real	[0, ∞)
FCarrier	carrier frequency	1000000.0	Hz	real	(0, ∞)
Power	unmodulated carrier power	0.01	W	real	[0, ∞)
Phase	unmodulated carrier phase, in degrees	0.0	deg	real	(-∞, ∞)
Sensitivity	modulator sensitivity in Hz/V	1000		real	(-∞, ∞)
FSignal	modulation signal frequency	1000	Hz	real	[0, ∞)
VPeak	peak modulation voltage level	1.0	V	real	(-∞, ∞)

Pin Outputs

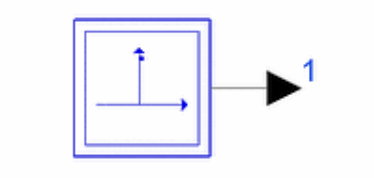
Pin	Name	Description	Signal Type
1	output	output signal	timed

Notes/Equations

1. For information regarding timed source signals, refer to *Timed Sources* (timed).
2. This source generates an RF (complex envelope) timed signal output at a characterization frequency of *FCarrier*. It is a frequency modulated carrier whose modulating signal is a single tone.
   Let
   v(t) = *VPeak* × cos(2π *FSignal t*)
   θ^{0} = π *Phase* / 180
   ω_{c} = 2π *FCenter*
   then

$$v_s(t) = 2sqrt(2ROutPower)e^{j\omega_c t + \theta_0 + 2\pi s\int_0^{\cdot}(v(\alpha)d\alpha)}e^{(-j)\omega_c t}$$

Impulse



Description: Baseband or RF impulse generator
Library: Timed, Sources
Class: TSDF_Impulse
Derived From: basesource

Parameters

Name	Description	Default	Unit	Type	Range
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
TStep	simulation time step	0.0001	sec	real	[0, ∞)
FCarrier	carrier frequency	0.0	Hz	real	[0, ∞)
Weight	impulse level	1.0	V	real	(-∞, ∞)
Period	impulse train period	0.001	sec	real	[2 * TStep), ∞)
Delay	time delay before turn-on	0.0	sec	real	[0, ∞)

Pin Outputs

Pin	Name	Description	Signal Type
1	output	output signal	timed

Notes/Equations

1. For information regarding timed source signals, refer to *Timed Sources* (timed).
2. This source generates an RF (complex envelope) timed signal output when FCarrier>0, or a baseband timed signal output when FCarrier=0.
   It is an impulse generator with a repetition period that can be set by the user.
   Let normalized time = ntime = fmod(t - Delay, Period)
   if ((ntime<0.0) or (ntime+TStep/2> Period))
   ntime = 0
   if ((t+TStep/2 ≥ Delay) and (ntime+TStep/2<TStep))
   v = Weight
   else
   v = 0
   if ROut = 0, then vs = v
   if ROut > 0, then vs = 2 × v
3. For example, a network placed in the schematic has the following z-transform transfer function:
   H(z) = z/(z - 0.5)
   IMP source shows the input impulse source and the resulting output signal of the network; TStep=1.0 sec, FCarrier=0, Weight=1, Delay=5 µsec, and Period=100 sec.

4. When using the Impulse source to evaluate the impulse response of a filter with a SpectrumAnalyzer or sink component, it is often desirable to set the impulse weight so that the spectrum measured is normalized to 0 dBv. For this application, the following values are recommended for Weight.
   For measurement of a baseband signal response, set Impulse source:

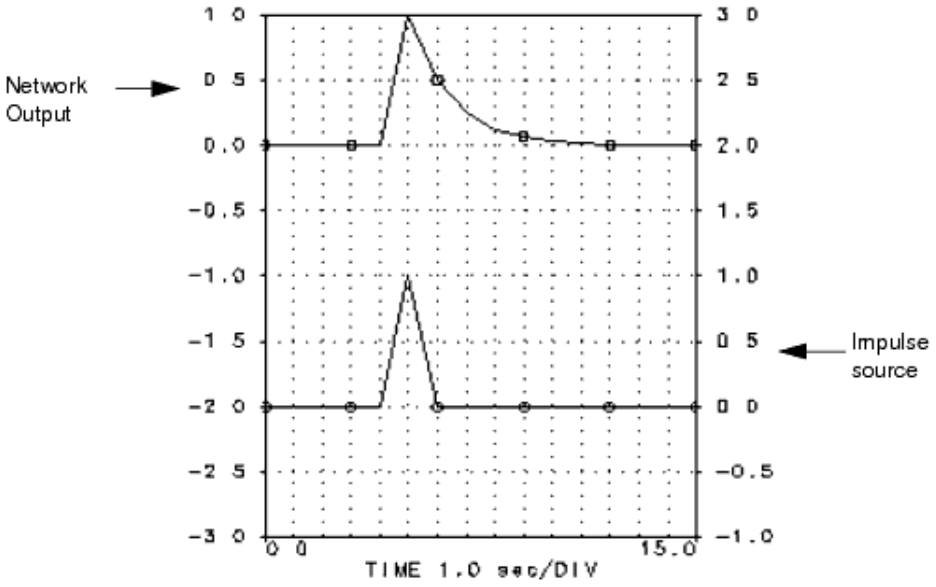
   $$Weight = \frac{Stop}{(2 \times TStep)}$$

   For measurement of an RF signal response, set Impulse source:

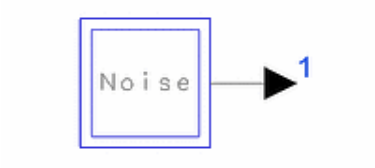
   $$Weight = \frac{Stop}{TStep}$$

   where TStep is the Impulse source value and Stop is the value associated with SpectrumAnalyzer.

IMP source



Noise



Description: Baseband or RF noise generator
Library: Timed, Sources
Class: TSDF_Noise
Derived From: basesource

Parameters

Name	Description	Default	Symbol	Unit	Type	Range
ROut	output resistance	DefaultROut		Ohm	real	(0, ∞)
RTemp	physical temperature in degrees C	DefaultRTemp		Celsius	real	[-273.15, ∞)
TStep	simulation time step	0.0001		sec	real	[0, ∞)
Type	Noise probability density or Cumulative distribution cdf function: Uniform PDF, Gaussian PDF, Rayleigh PDF, LogNormal PDF, Exponential PDF, Weibull PDF, ChiSquared PDF, Gamma PDF, Beta PDF, F PDF, Binomial CDF, Poisson CDF	Gaussian PDF			enum	
FCarrier	Carrier frequency	0.0		Hz	real	[0, ∞)
VA	Voltage value dependent on Type	1.0	A	V	real	
VB	Voltage value dependent on Type	1.0	B	V	real	
Delay	Time delay before turn on	0.0		sec	real	[0, ∞)
DurationTime	Time duration for noise on	1	D	sec	real	(0, ∞)
RepetitionInterval	Repetition interval for noise to turn back on	.002		sec	real	[D, ∞)

Pin Outputs

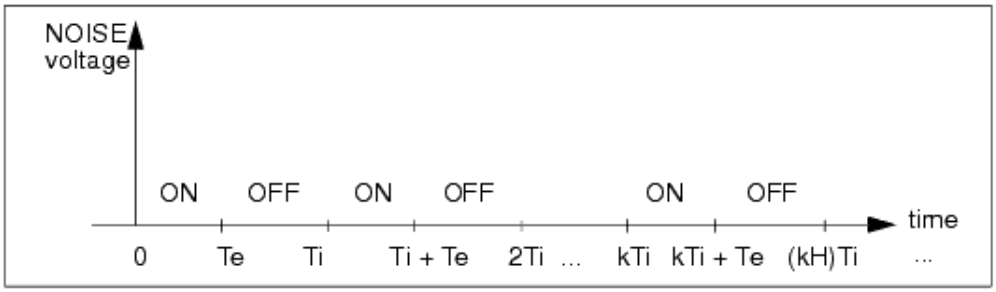
Pin	Name	Description	Signal Type
1	output	output signal	timed

Notes/Equations

1. For information regarding timed source signals, refer to *Timed Sources* (timed).
2. This source generates an RF (complex envelope) timed signal output when FCarrier > 0, or a baseband timed signal output when FCarrier = 0. It is a noise generator with user-settable probability density function (pdf) or cumulative distribution function (cdf).
   Let
   vi = noise as a function of time with the characteristics specified.
   If FCarrier=0, let
   vq = 0

else, let
vq = 2nd noise as a function of time with the characteristics specified
Let normalized time = ntime = fmod(t−Delay, RepetitionInterval)
if ((t+TStep/2<Delay) or (ntime+TStep/2.> DurationTime))
vs = 0
vs = 2(vi + j×vq)

3. The Type parameter must be specified before the VA and VB parameters because the pdf/cdf type determines the available voltage types in VA and VB. Noise Types shows the probability density function (pdf) or cumulative distribution function (cdf) types for the Type parameter.
4. In the following
   Let
   Te = DurationTime
   Ti = RepetitionInterval
   The parameters Te and Ti can be used to generate a pulsed noise source:

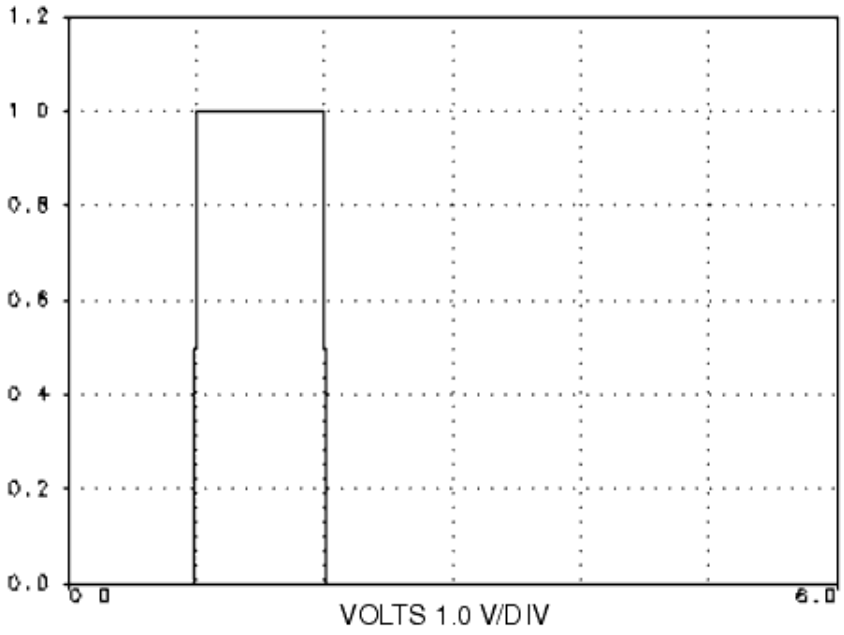


   If a continuous noise source is required, Te and Ti must be set equal.
5. Noise contributed from Noise instance is an independent noise process; noise is dependent on the value of the DefaultSeed in the DF controller; for more information, refer to DF (data flow) controller documentation. When DefaultSeed=0, the noise generated for each simulation is different. When DefaultSeed>0, the noise generated for each simulation, though random, has the same initial seed starting condition and thus results in reproducible noise for each simulation
6. Noise Types and Uniform PDF Value Versus Voltage with A=1 and B=2 through Poisson CDF Value Versus Voltage with A= 2 and B=0.1 show each of the available Noise types.

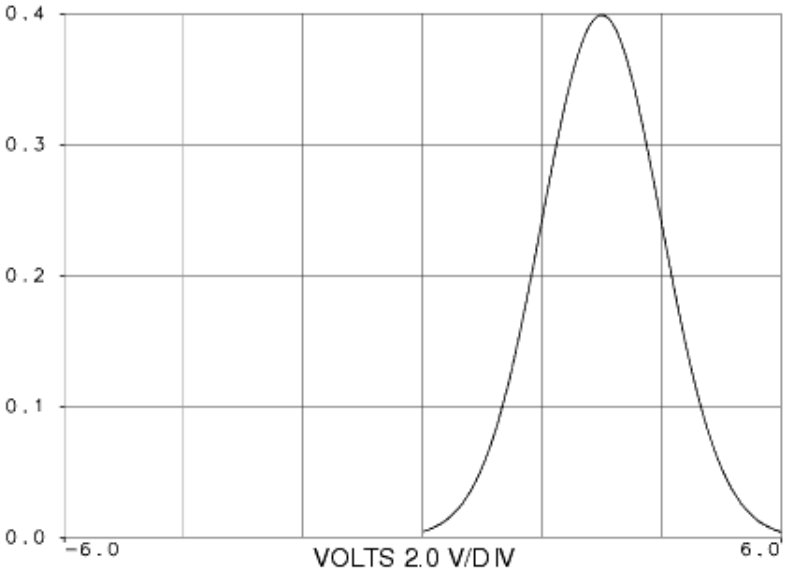
Noise Types

Noise Type	PDF f(v) or CDF F(v)
Gaussian PDF	$f(v) = \frac{1}{\sqrt{2\pi B^2}} e^{-\frac{(v-A)^2}{2B^2}}$
Uniform PDF	$f(v) = \begin{cases} \frac{1}{B-A} & \text{if } A \le v \le B \\ 0 & \text{otherwise} \end{cases}$
Rayleigh PDF	$f(v) = 2\ln(2)\frac{v}{A^2} e^{-\ln(2)\left(\frac{v}{A}\right)^2} U(v)$
Log-normal PDF	$f(v) = \begin{cases} \frac{1}{\sqrt{2\pi B^2 v^2}} e^{-\frac{1}{2}\left(\frac{\ln(v)-A}{B}\right)^2} & \text{if } v > 0 \\ 0 & \text{otherwise} \end{cases}$
Exponential PDF	$f(v) = \frac{\ln 2}{A} e^{\frac{-v \ln 2}{A}} U(v)$
Weibull PDF	$f(v) = \frac{2B \ln 2}{A^{2B}} v^{2B-1} e^{-\ln 2\left(\frac{v}{A}\right)^{2B}} U(v)$
Chi-squared PDF	$f(v) = \frac{v^{A-1} e^{-v/2}}{2^A \Gamma(A)} U(v)$
Gamma PDF	$f(v) = \frac{A^B}{\Gamma(B)} e^{-Av} v^{B-1} U(v)$
Beta PDF	$f(v) = \begin{cases} \frac{\Gamma(A+B)}{\Gamma(A)\Gamma(B)} v^{A-1}(1-v)^{B-1} & \text{for } 0 < v < 1 \\ 0 & \text{otherwise} \end{cases}$
F PDF	$f(v) = A^{A/2} B^{B/2} \frac{\Gamma\left(\frac{A+B}{2}\right)}{\Gamma(A/2)\Gamma(B/2)} \frac{v^{A/2-1}}{(A+Bv)^{(A+B)/2}} U(v)$
Binomial CDF	$F(v) = \sum_{k=0}^{v} \binom{A}{k} B^k (1-B)^{(v-k)}$
Poisson CDF	$F(v) = \sum_{k=0}^{v} \frac{A^k}{k!} e^{-A}$

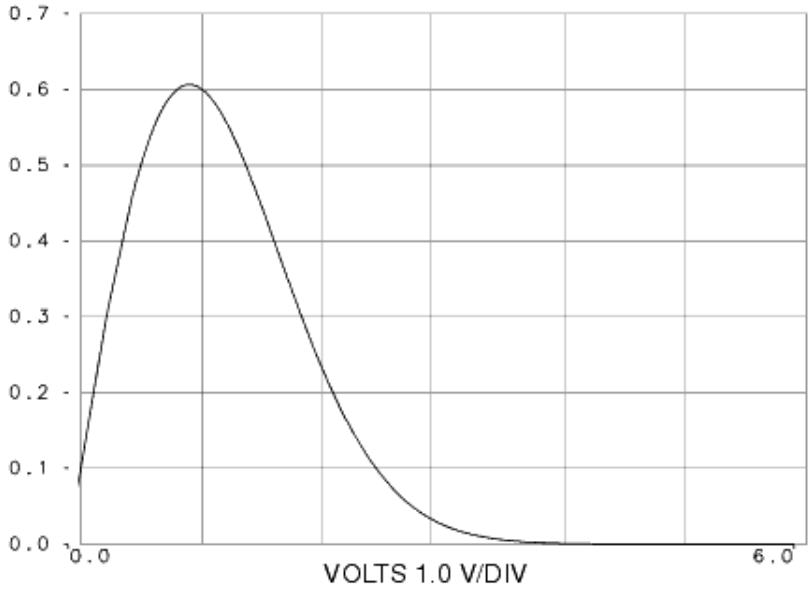
Uniform PDF Value Versus Voltage with A=1 and B=2



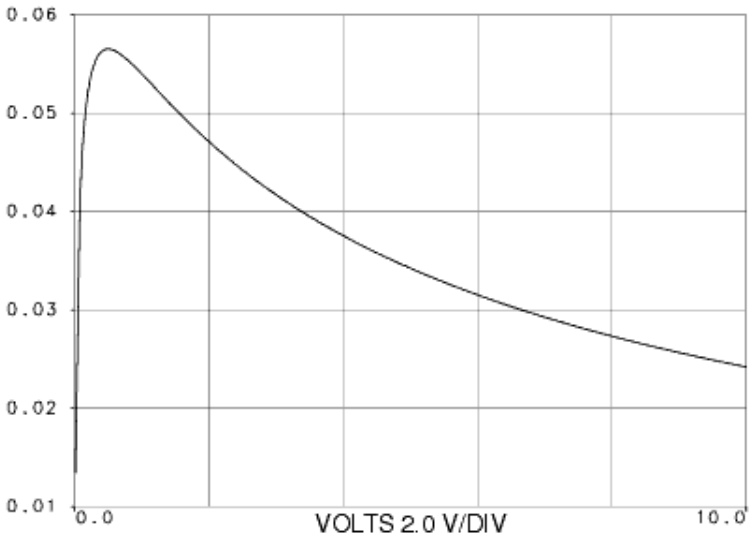
Gaussian PDF Value Versus Voltage with A=3 and B=1



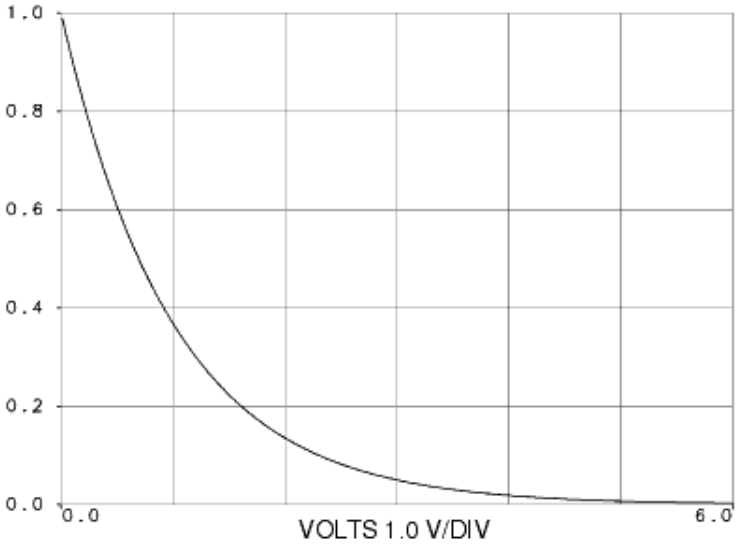
Rayleigh PDF Value Versus Voltage with A = sqrt(2 *ln* (2))



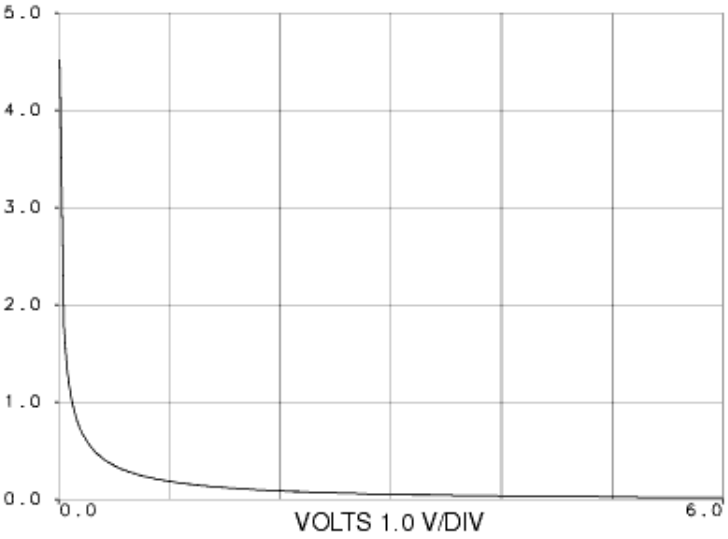
Log-Normal PDF Value Versus Voltage with A=2 and B=1



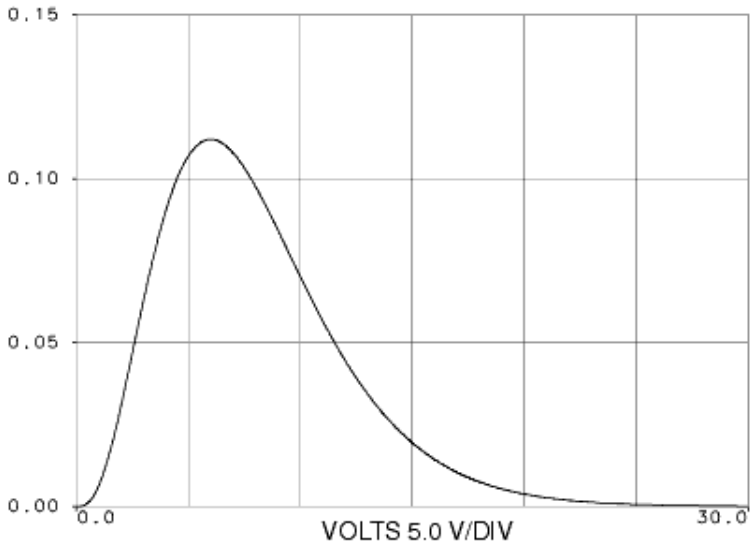
Exponential PDF Value Versus Voltage with A=ln2



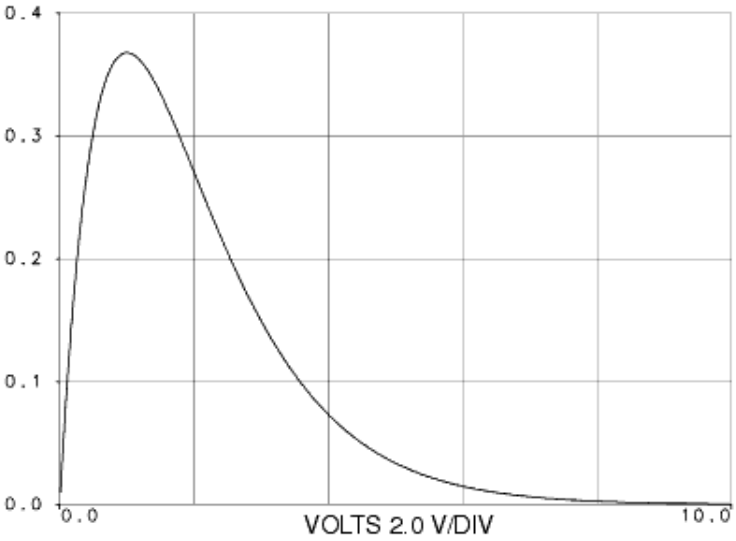
Weibull PDF Value Versus Voltage with A=0.48 and B=0.25



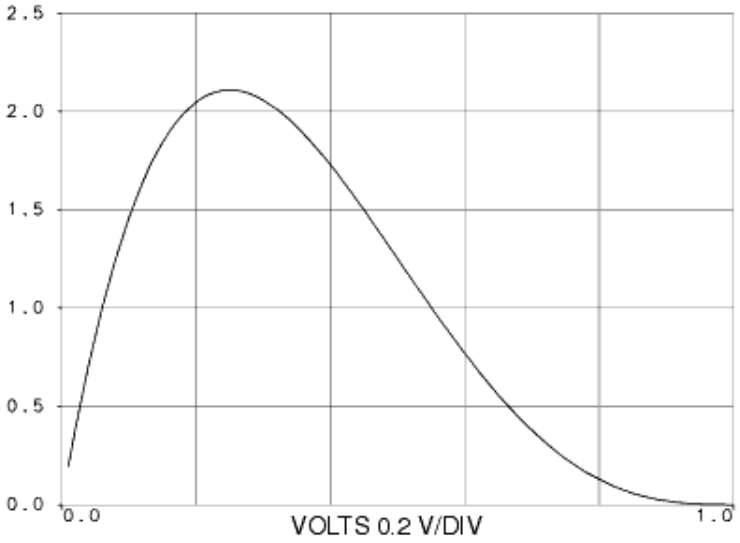
Chi-Squared PDF Value Versus Voltage with A=4



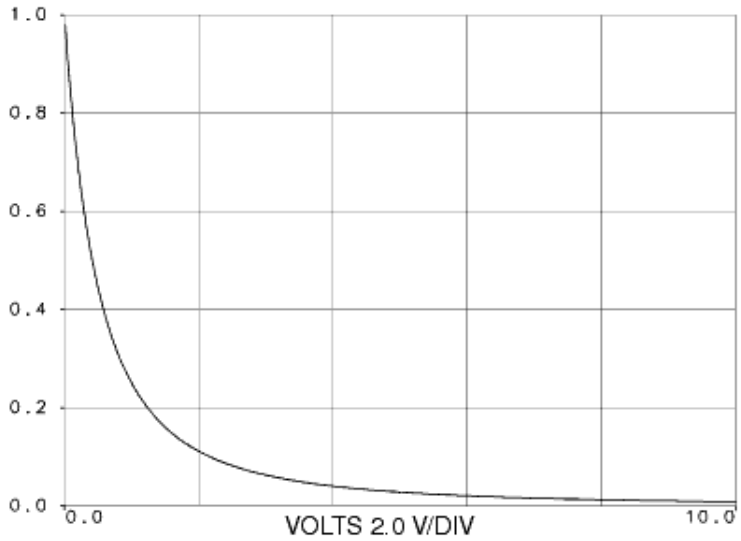
Gamma PDF Value Versus Voltage with A=1 and B=2



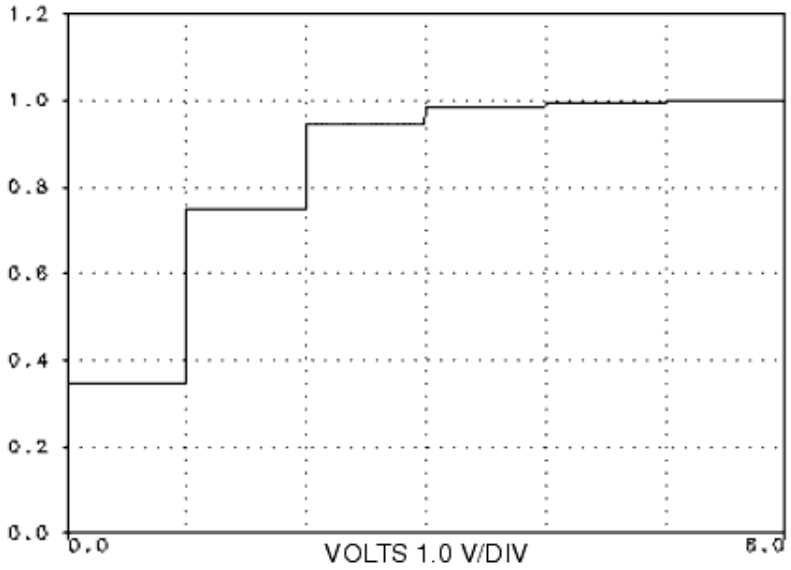
Beta PDF Value Versus Voltage with A=2 and B=4



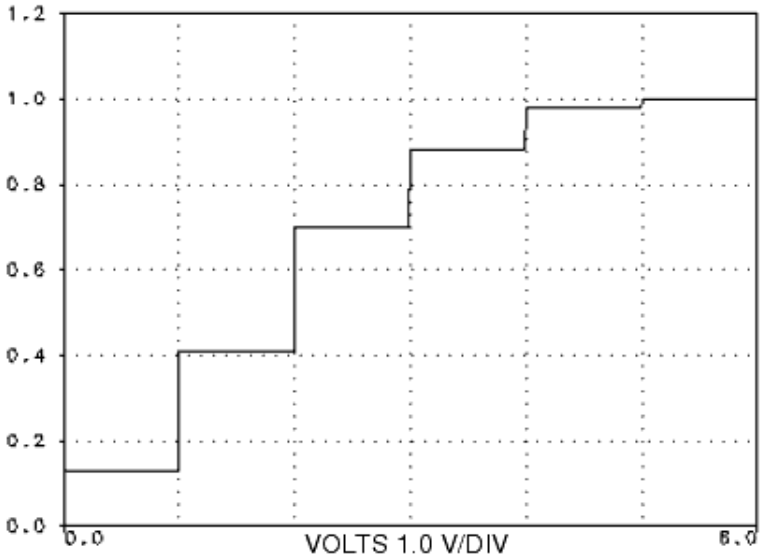
F PDF Value Versus Voltage with A=2 and B=2



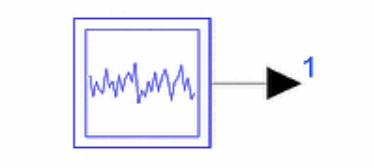
Binomial CDF Value Versus Voltage with A=10 and B=0.1



Poisson CDF Value Versus Voltage with A= 2 and B=0.1



NoiseFMask



Description: Noise generator with frequency domain mask specification
Library: Timed, Sources
Class: TSDF_NoiseFMask
Derived From: basePowerSource

Parameters

Name	Description	Default	Unit	Type	Range
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
TStep	simulation time step	0.0001	sec	real	[0, ∞)
NoiseMask	noise specification defined with pairs of values for frequency (Hz), noise level (dBm/Hz)			real array	
ResBW	resolution frequency bandwidth for noise spectrum	1.0 MHz	Hz	real	(0, ∞)†

† ResBW must be smaller than the simulation bandwidth (1/TStep) and smaller than the noise bandwidth (defined by the smallest and largest frequency in the NoiseMask array).

Pin Outputs

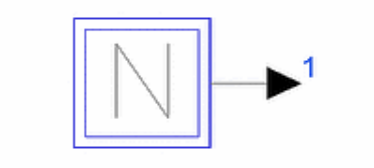
Pin	Name	Description	Signal Type
1	output	output signal	timed

Notes/Equations

1. For information regarding timed source signals, refer to *Timed Sources* (timed).
2. NoiseFMask generates an RF (complex envelope) noise signal with a power spectral density that follows the mask specified in the NoiseMask parameter.
3. The NoiseMask parameter is an array of values that defines the mask. The values are interpreted as pairs of frequency in Hz and noise power level in dBm/Hz. The characterization frequency of the output signal is equal to $(f_{min} + f_{max})/2$, where f_{min} (f_{max}) is the lowest (highest) frequency specified in the NoiseMask parameter. The simulation time step should be small enough to support a bandwidth of ($f_{max} - f_{min}$).
4. The output signal is modeled as a sum of tones. The number of tones is ($f_{max} - f_{min}$) / ResBW. The higher the number of tones used, the more accurately the output signal will be modeled. However, using more tones will increase simulation time.

The frequency of the i^{th} tone is randomly selected using a uniform distribution in the interval $[f_{min} + i \times ResBW, f_{min} + (i+1) \times ResBw)$. The amplitude of the i^{th} tone is randomly selected using a normal distribution centered around the amplitude value that corresponds to the power level specified in the mask (power levels for tones whose frequency is not listed in the NoiseMask parameter are calculated using interpolation). The initial phase of each tone is randomly selected using a uniform distribution in the interval $[0, 2\pi)$.

N_Tones



Description: RF tones generator
Library: Timed, Sources
Class: TSDF_N_Tones
Derived From: basesource

Parameters

Name	Description	Default	Unit	Type	Range
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
TStep	simulation time step	0.0001	sec	real	[0, ∞)
Frequency1	first RF tone frequency	999999	Hz	real	(0, ∞)
Power1	first RF tone carrier power	.010	W	real	[0, ∞)
Phase1	first RF tone carrier phase in degrees	0.0	deg	real	(-∞, ∞)
AdditionalTones	list of additional RF tones defined with triple values for frequency (Hz), power (watts), phase (degrees)			real array	
RandomPhase	set phase of RF tones to random uniformly distributed value between -PI and +PI: No, Yes	No		enum	
PhaseNoiseData	phase noise specification defined with pairs of values for offset frequency (Hz), signal sideband pnase noise level (dBc/Hz)			real array	
PN_Type	Phase noise model type with random or fixed offset freq spacing and amplitude: Random PN, Fixed freq offset, Fixed freq offset and amplitude	Random PN		enum	

Pin Outputs

Pin	Name	Description	Signal Type
1	output	output signal	timed

Notes/Equations

1. For information regarding timed source signals, refer to *Timed Sources* (timed).
2. N_Tones generates an RF (complex envelope) timed signal output. It generates an N-tone RF signal, where each tone is specified by its frequency, power, and phase. The phase can be specified as a random function. Additionally, each tone may be colored with a user-specified phase noise characteristic.
3. The frequency, power, and phase of the first tone are defined by the Frequency1, Power1, and Phase1 parameters, respectively. Additional tones can be defined in the AdditionalTones list. This list contains triple value of frequency (Hz), power (watts), and phase (degrees).

```
AdditionalTones = "1.01e6, 0.001, 0, 1.02e6, 0.0015, 90.0"
```

As an alternative from listing these additional tones, this dataset may be contained in a text file and referenced by name as follows:

```
AdditionalTones = "<mytonelist.re"
```

The *mytonelist.re* file must be located in the current workspace data subdirectory. If not in the data subdirectory, then the file name must include the full directory path as the prefix to the file name. The contents of this file simply triples values for each tone, where the number separator can be a comma, space, tab, or new line:
1.01e6, 0.001, 0
1.02e6, 0.0015, 90.0
For details on setting and using arrays of data for parameter values, see *Array Parameters* (ptolemy) in the *ADS Ptolemy Simulation* (ptolemy) documentation.

4. If RandomPhase=Yes, then the phase in the list of tones is ignored and a random phase between −180 and +180 is used.
   When RandomPhase=Yes, the phase for each tone for each N_Tones instance is an independent random process. This random phase is dependent on the value of DefaultSeed in the DF (data flow) controller. When DefaultSeed = 0, the random phase generated for each simulation is different. When DefaultSeed > 0, the random phase generated for each simulation, though random, has the same initial seed starting condition and thus results in reproducible random phase for each simulation.
5. A phase noise characteristic is specified in the PhaseNoiseData list. This list contains double values of offset frequency (Hz) and single sideband relative power level (dBc/Hz).

```
PhaseNoiseData = "100, -50,  1000, -70, 1e5, -90",  or with alternate syntax @{ 100, -50,
1K, -70, 100K, -90}
```

This second curly array entry syntax supports use of variables, variable expressions, simulator expressions, and multiplier symbols (p, n, u, m, k, M, G, ...).
As an alternative to listing this phase noise, this dataset can be contained in a text file and referenced by name as follows:

```
PhaseNoiseData = "<myphasenoise.re"
```

The *myphasenoise.re* file must be located in the current workspace data subdirectory. If not in the data subdirectory, then the file name must include the full directory path as the prefix to the file name. The contents of this file is simply the double values for each tone, where the number separator can be a comma, space, tab, or new line:
100, -50
1000, -70
1e5, -90
For details on setting and using arrays of data for parameter values, see *Array Parameters* (ptolemy) in the *ADS Ptolemy Simulation* (ptolemy) documentation.
The phase noise characteristic defined from this list describes a frequency domain specification for phase noise. Interpolation is applied between these frequency domain points as needed to give a full time domain simulation definition for this phase noise.

Each signal tone is defined by its complex envelope at the source signal carrier frequency FCarrier. This value is set as follows:
FCarrier = (MaxFreq + MinFreq) /2
where MaxFreq (MinFreq) is the maximum (minimum) frequency of all signal tones.
The source output signal is the summation of all signal tones characterized at this carrier frequency.

Each signal tone is modeled by the expression

$$V(t) = V_p \times \cos\left(2\pi f_c t + \left(\sum_{i=1}^{N} \eta_i \times \sin(2\pi f_i t + \phi_i)\right) + \phi\right)$$

where

$$V_p = \sqrt{2 \times P \times ROut}$$

(P is the tone power and ROut is the source output resistance)

f_c is the tone frequency

φ is the tone phase

and the summation term represents the phase noise.

Phase noise is modeled as a sum of tones that modulates the phase of the main tone. Let $f_{offsetMax}$ ($f_{offsetMin}$) be the maximum (minimum) offset frequency specified in PhaseNoiseData. Then the number of tones N used to model phase noise is equal to $f_{offsetMax} / f_{offsetMin}$.

Let P_i be the phase noise power at frequency offset f_i. The phase (φ_i), frequency (f_i), and amplitude (or modulation index) (η_i) of each tone is given by:

When PN_Type = Fixed freq offset and amplitude

φ_i is a random variable uniformly distributed in [0, 2π)

$$f_i = i \times f_{offsetMin}$$

$$\eta_i = \sqrt{2 \times P_i \times ROut}$$

When PN_Type = Fixed freq offset

φ_i is a random variable uniformly distributed in [0, 2π)

$$f_i = i \times f_{offsetMin}$$

η_i is a random variable with a Rayleigh distribution and mean value

$$\sqrt{2 \times P_i \times ROut}$$

When PN_Type = Random PN

φ_i is a random variable uniformly distributed in [0, 2π)

f_i is a random variable uniformly distributed in

$$[(i - 0.5) \times f_{offsetMin}, (i + 0.5) \times f_{offsetMin})$$

η_i is a random variable with a Rayleigh distribution and mean value

$$\sqrt{2 \times P_i \times ROut}$$

For the first two cases *Fixed freq offset and amplitude* and *Fixed freq offset*, phase noise is a sum of tones whose frequencies are integer multiples of the same frequency ($f_{offsetMin}$). Therefore, phase noise will be periodic with period 1 / $f_{offsetMin}$ and all the phase noise signal power will be located at the discrete frequencies that are integer multiples of $f_{offsetMin}$. When a spectrum analysis is performed on this signal and the resolution bandwidth is equal to $f_{offsetMin}$ / M, where M is an integer, the spectrum will have spectral nulls (e.g. -250 dBm) at all frequencies that are not an integer multiple of $f_{offsetMin}$ (see Phase Noise Modeling Example). The integrated power in a bandwidth of $f_{offsetMin}$ will still be what one expects based on the phase noise data specification but it will all be concentrated at one frequency (the one that is an integer multiple of $f_{offsetMin}$).

For better phase noise modeling it is recommended that simulations be performed with PN_Type set to Random PN (default). The other values for PN_Type can be used to demonstrate/understand the phase noise modeling algorithm and are not recommended for use in practical simulations.
The single sideband phase noise in dBc/Hz is

$$\Im(f_i) = 10 \times \log\left(\frac{\eta_{rms}{}^2(f_i)}{2}\right) \text{ at offset frequency } f_i$$

where
$\eta_{rms}(f_i)$ = root mean square of modulation index η_i at offset frequency f_i.
The modulation index is related to the signal power and single sideband phase noise power as follows:

$$\frac{P_{ssb}(f_i)}{P_c} = \frac{\eta_{rms}{}^2(f_i)}{2}$$

where
P_c = signal power
$P_{ssb}(f_i)$ = signal sideband power at offset frequency *f* i
Therefore, single sideband phase noise in dBc/Hz can also be expressed as:

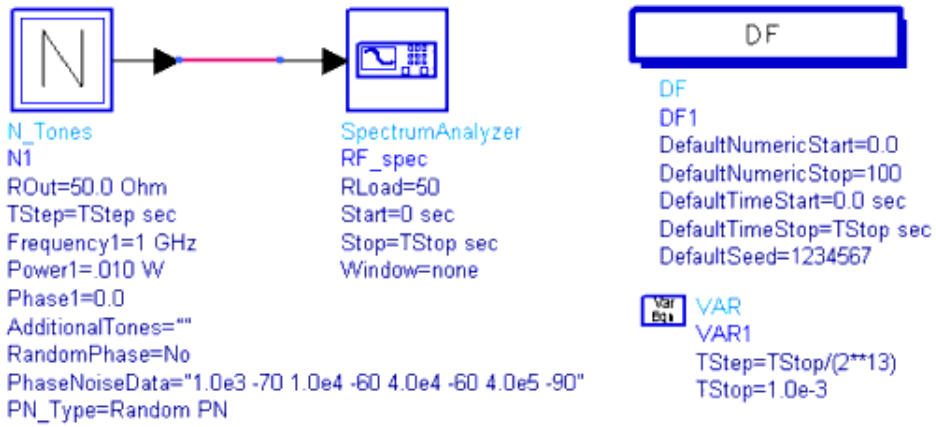
$$\Im(f_i) \text{ in dBc/Hz} = P_{ssb}(f_i)dBm - ResBWdB - P_c dBm$$

where
$P_{ssb}(f_i)dBm$ = simulated single sideband phase noise power per simulation frequency resolution bandwidth ResBW
ResBWdB = factor for frequency resolution bandwidth ResBW used during simulation = 10 × log10(ResBW)
P_c *dBm* = signal power in dBm

Note
If the phase noise settings result in the summed phase noise higher than *tone power* (dBm) - 10 dB, a warning message will be generated:
Phase noise violates small signal modeling requirement; phase noise power exceeds Tone_Power (dBm) - 10 dB.
However, the simulation will continue with the current parameter settings.

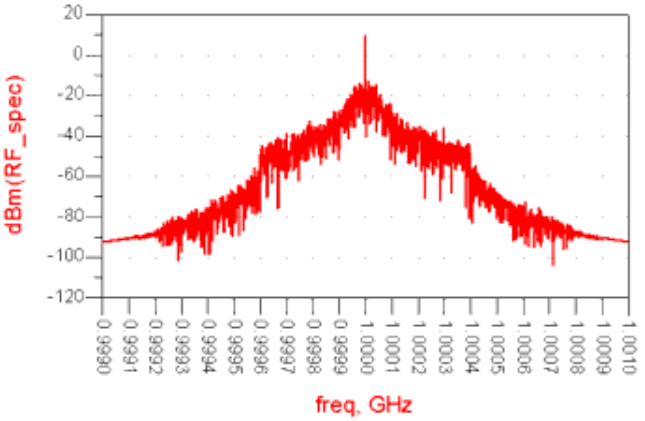
A phase noise modeling example is demonstrated here. Consider the simple design shown in Phase Noise Modeling Example. An N_Tones source is used to generate a 1 GHz tone at a power level of 10 dBm into 50 Ohms. The tone is colored with phase noise, whose frequency specification is defined in the PhaseNoiseData parameter. For this example, $f_{offsetMin}$ is 1 kHz and $f_{offsetMax}$ is 400 kHz.

Phase Noise Modeling Example



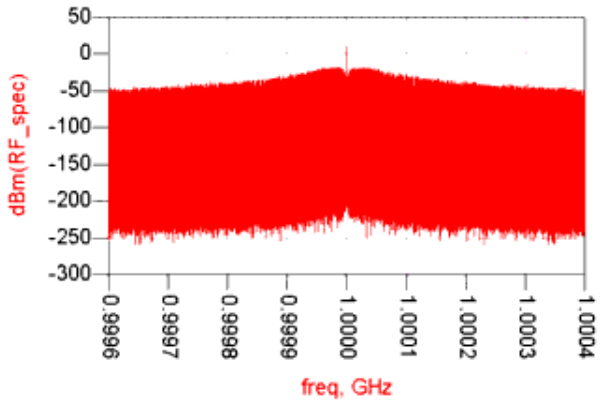
The simulation time step is set to (1 / 2^{13}) msec, which is small enough to resolve the maximum phase noise frequency offset data point at 400 kHz. The simulation stop time is set to 1 msec, which is large enough to resolve the lowest phase noise frequency offset data point at 1 kHz.
The spectrum of the signal at the output of N_Tones is measured using a SpectrumAnalyzer component. The spectrum is shown in Signal Spectrum with PN_Type=Random PN and TStop=1 msec. The resolution bandwidth of the spectrum measurement is 1 / (Stop - Start) = 1 / (1 msec) = 1 kHz. This means that each spectral tone displayed will be at multiples of 1 kHz from the carrier frequency of 1 GHz. For noise power integrated over a 1 kHz bandwidth, the power would be 30 dB (=10 × log10(1000)) more than that in a 1 Hz bandwidth.

Signal Spectrum with PN_Type= *Random PN* and TStop=1 msec



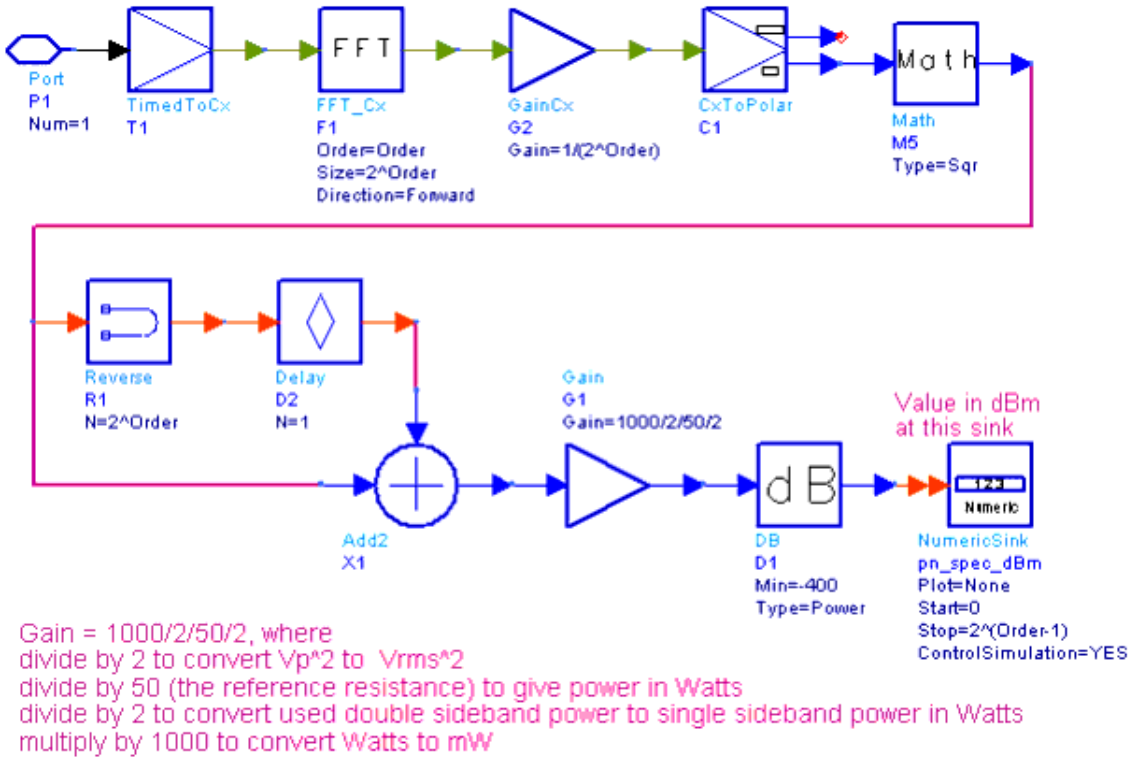
Signal Spectrum with PN_Type=Fixed freq offset and amplitude and TStop=2 msec shows the signal spectrum when PN_Type= *Fixed freq offset and amplitude* and TStop=2 msec. The resolution bandwidth of the spectrum measurement is 1 / (Stop - Start) = 0.5 kHz. As explained in *note 8*, all phase noise signal power will be located at frequencies that are integer multiples of $f_{offsetMin}$ = 1 kHz and the spectrum values at frequencies (M+0.5) kHz (where M is an integer) will be practically 0.

Signal Spectrum with PN_Type= *Fixed freq offset and amplitude* and TStop=2 msec



To view the phase noise spectrum versus spectral tone offset index, a signal processing network can be created as shown in Signal Processing Network to Calculate Phase Noise Spectrum vs. Spectral Tone Offset Index to perform FFT on the collected RF complex time domain waveform. Summed powers in the upper and lower sidebands are averaged and results are converted into dBm to obtain the single sideband phase noise power in dBm per simulation frequency resolution bandwidth versus offset spectral tone. This resultant single sideband phase noise spectrum can be displayed versus spectral tone offset index as shown in Phase Noise Spectrum with PN_Type=Random PN.

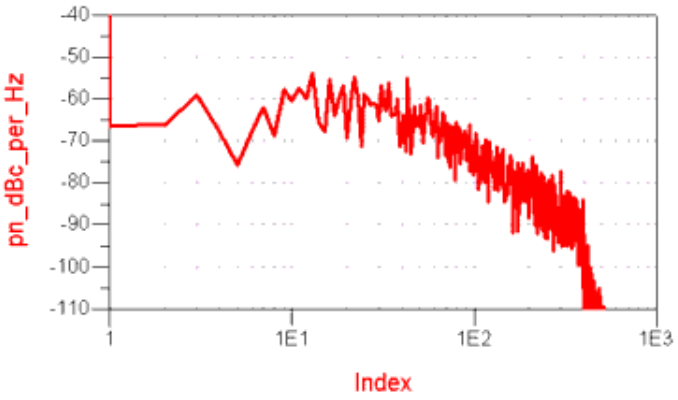
Signal Processing Network to Calculate Phase Noise Spectrum vs. Spectral Tone Offset Index



In Phase Noise Spectrum with PN_Type=Random PN, 1 kHz offset occurs at index 1, 10 kHz offset occurs at index 10, 40 kHz offset occurs at index 40, and 400 kHz offset occurs at index 400. As can be seen, this figure agrees with the PhaseNoiseData specified.
The phase noise data displayed in Phase Noise Spectrum with PN_Type=Random PN was generated using PN_Type= *Random PN* . In Phase Noise Spectrum with PN_Type=Fixed freq offset and amplitude, phase noise is displayed with PN_Type= *Fixed freq offset and amplitude* . As can be seen, Phase Noise Spectrum with PN_Type=Fixed freq offset and amplitude agrees much better (compared to Phase Noise Spectrum with PN_Type=Random PN) with the PhaseNoiseData specified, since in this case there is no randomness in the values of f_i and η_i .

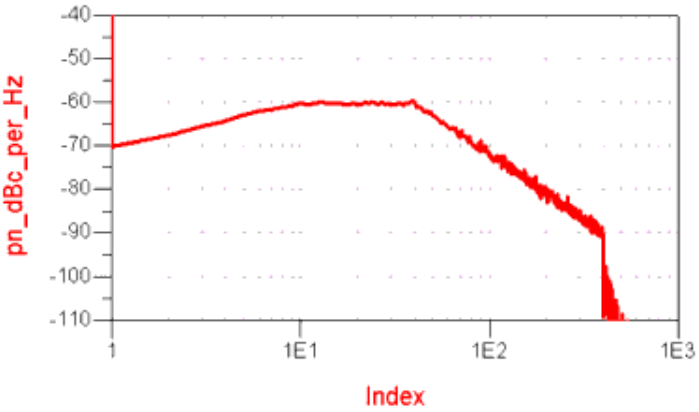
The results of Phase Noise Spectrum with PN_Type=Random PN and Phase Noise Spectrum with PN_Type=Fixed freq offset and amplitude were obtained by setting the Order parameter (of the signal processing network Signal Processing Network to Calculate Phase Noise Spectrum vs. Spectral Tone Offset Index) to 13.

Phase Noise Spectrum with PN_Type= *Random PN*



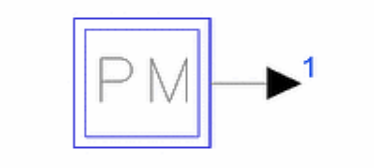
Eqn pn_dBc_per_Hz = pn_spec_dBm - 10 - 30

Phase Noise Spectrum with PN_Type= *Fixed freq offset and amplitude*



Eqn pn_dBc_per_Hz = pn_spec_dBm - 10 - 30

PM



Description: Phase modulated carrier with single modulating tone
Library: Timed, Sources
Class: TSDF_PM
Derived From: basesource
C++ Code: See *doc/sp_items/TSDF_PM.html* under your installation directory.

Parameters

Name	Description	Default	Unit	Type	Range
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
TStep	simulation time step	0.0001	sec	real	[0, ∞)
FCarrier	carrier frequency	1000000.0	Hz	real	(0, ∞)
Power	unmodulated carrier power	0.01	W	real	[0, ∞)
Phase	unmodulated carrier phase, in degrees	0.0	deg	real	(-∞, ∞)
Sensitivity	modulator sensitivity in radians/V	pi/4		real	(-∞, ∞)
FSignal	modulation signal frequency	1000	Hz	real	[0, ∞)
VPeak	peak modulation voltage level	1.0	V	real	(-∞, ∞)

Pin Outputs

Pin	Name	Description	Signal Type
1	output	output signal	timed

Notes/Equations

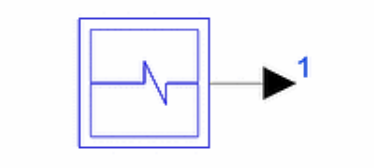
1. For information regarding timed source signals, refer to *Timed Sources* (timed).
2. This source generates an RF (complex envelope) timed signal output at a characterization frequency of *FCarrier*. It is a phase modulated carrier whose modulating signal is a single tone.

$$vs = 2 \times \sqrt{2 \times Power \times ROut} \times e^{j\theta}$$

where

$$\theta = \frac{\pi \times Phase}{180} + Sensitivity \times VPeak \times \cos(2 \times \pi \times FSignal \times t)$$

Pulse



Description: Baseband or RF pulse generator
Library: Timed, Sources
Class: TSDF_Pulse
Derived From: basesource
C++ Code: See *doc/sp_items/TSDF_Pulse.html* under your installation directory.

Parameters

Name	Description	Default	Symbol	Unit	Type	Range
ROut	output resistance	DefaultROut		Ohm	real	(0, ∞)
RTemp	physical temperature in degrees C	DefaultRTemp		Celsius	real	[-273.15, ∞)
TStep	simulation time step	0.0001		sec	real	[0, ∞)
FCarrier	carrier frequency	0.0		Hz	real	[0, ∞)
VStart	start voltage level	0.0		V	real	(-∞, ∞)
VPlateau	plateau voltage level	1.0		V	real	(-∞, ∞)
VStop	modulator reference voltage level	0.0		V	real	(-∞, ∞)
Delay	time delay before turn-on	0.0005		sec	real	[0, ∞)
LeadingEdgeTime	time interval from VStart to VPlateau	0.0002	LE	sec	real	[0, ∞)†
PlateauTime	time interval at VPlateau level	0.0006	PL	sec	real	[newpro:0, ∞)†
TrailingEdgeTime	time interval from VPlateau to VStop	0.0002	TE	sec	real	[0, ∞)†
RepetitionInterval	repetition time interval	0.001		sec	real	[LE + PL + TE, ∞)

† This parameter must satisfy the condition
if (LE=0 and TE=0)
0 < TStep < PL
else
0 < TStep < min(PL, LE, TE)

Pin Outputs

Pin	Name	Description	Signal Type
1	output	output signal	timed

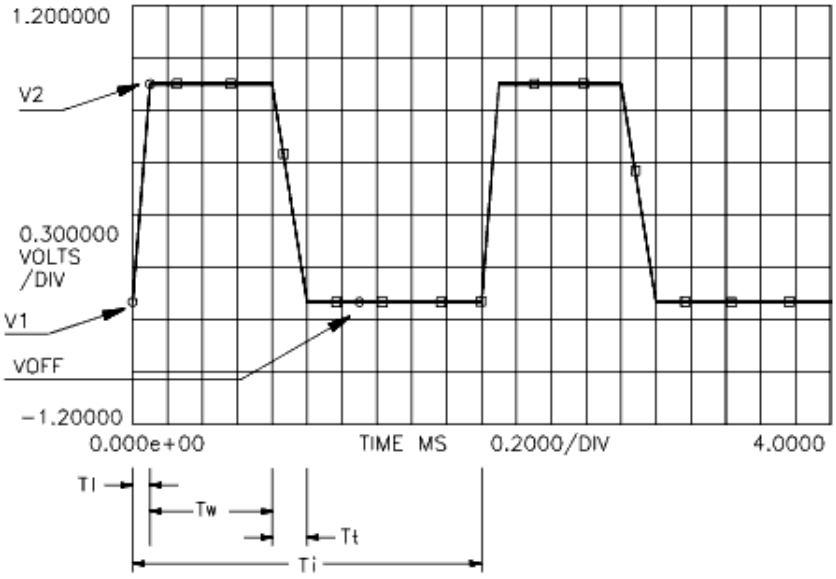
Notes/Equations

1. For information regarding timed source signals, refer to *Timed Sources* (timed).
2. This source generates an RF (complex envelope) timed signal output when FCarrier >

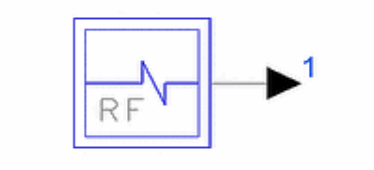
0, or a baseband timed signal output when FCarrier = 0. It is a pulse generator with settable leading and trailing edge times, plateau time, and repetition interval.
Let
v = basic pulse waveform
If ROut =0, then vs = v
If ROut > 0, then vs = 2×v

3. Baseband Pulse Signal illustrates a baseband pulse signal.
Let
T1= LeadingEdgeTime
Tw = PlateauTime
Tt = TrailingEdgeTime
Ti = RepetitionInterval
V1 = VStart
V2 = VPlateau
VStop = VOFF

Baseband Pulse Signal



PulseRF



Description: Pulsed RF signal generator
Library: Timed, Sources
Class: TSDF_PulseRF
Derived From: basesource
C++ Code: See *doc/sp_items/TSDF_PulseRF.html* under your installation directory.

Parameters

Name	Description	Default	Symbol	Unit	Type	Range
ROut	output resistance	DefaultROut		Ohm	real	(0, ∞)
RTemp	physical temperature in degrees C	DefaultRTemp		Celsius	real	[-273.15, ∞)
TStep	simulation time step	0.0001		sec	real	[0, ∞)
FCarrier	carrier frequency	1000000.0		Hz	real	(0, ∞)
PeakPower	peak power level	0.01		W	real	[0, ∞)
Phase	unmodulated carrier phase, in degrees	0.0		deg	real	(-∞, ∞)
PulsePeriod	pulse period	0.002		sec	real	[(RT+FT) *0.625+PW, ∞)
PulseWidth	pulse width (between 50% levels)	0.001	PW	sec	real	(0, ∞)†
Delay	time delay before turn on	0.0		sec	real	[0, ∞)
OnOffRatio	on/off ratio, in dB	200			real	(-∞, ∞)
RiseTime	pulse rise time (between 10% and 90% levels)	0.00016	RT	sec	real	[newpro:0, ∞)†
FallTime	pulse fall time (between 90% and 10% levels)	0.00016	FT	sec	real	[0, ∞)†

† This parameter must satisfy the condition
if (RT=0 and FT=0)
0 < TStep < PW
else
0 < TStep < min(PW, RT, FT)

Pin Outputs

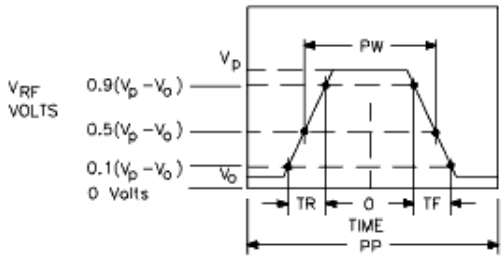
Pin	Name	Description	Signal Type
1	output	output signal	timed

Notes/Equations

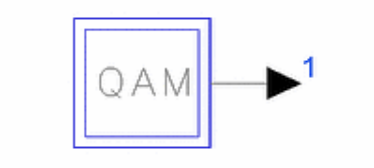
1. For information regarding timed source signals, refer to *Timed Sources* (timed).

2. This source generates an RF (complex envelope) timed signal output at a characterization frequency of *FCarrier*. It is a pulse modulated RF signal with settable on/off ratio, pulse period, and rise and fall times. RF Pulse Envelope illustrates an RF pulse envelope.
3. The output of this source is represented by its inphase and quadrature components at the carrier frequency.
   Let
   PW = PulseWidth
   TR = RiseTime
   TF = FallTime
   PP = PulsePeriod
   V_p = volts at PeakPower
   V_o = volts at PeakPower (dBm) minus the OnOffRatio (dB)

RF Pulse Envelope



QAM



Description: Quadrature amplitude modulated carrier with single I and Q modulating tones
Library: Timed, Sources
Class: TSDF_QAM
Derived From: basesource
C++ Code: See *doc/sp_items/TSDF_QAM.html* under your installation directory.

Parameters

Name	Description	Default	Unit	Type	Range
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
TStep	simulation time step	0.0001	sec	real	[0, ∞)
FCarrier	carrier frequency	1000000.0	Hz	real	(0, ∞)
Power	unmodulated carrier power	0.01	W	real	[0, ∞)
Phase	unmodulated carrier phase, in degrees	0.0	deg	real	(-∞, ∞)
VRef	modulator voltage reference level	1	V	real	(0, ∞)
FSignalI	I envelope modulation frequency	1000	Hz	real	(0, ∞)
VPeakI	I envelope peak modulation voltage level	1.0	V	real	(-∞, ∞)
FSignalQ	Q envelope modulation frequency	2000	Hz	real	(0, ∞)
VPeakQ	Q envelope peak modulation voltage level	1.0	V	real	(-∞, ∞)

Pin Outputs

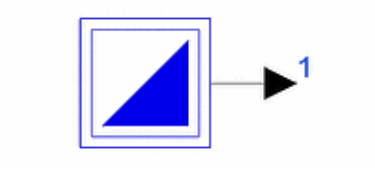
Pin	Name	Description	Signal Type
1	output	output signal	timed

Notes/Equations

1. For information regarding timed source signals, refer to *Timed Sources* (timed).
2. This source generates an RF (complex envelope) timed signal output at a characterization frequency of *FCarrier*. It is a quadrature amplitude modulated carrier whose I and Q modulating signals are single tones.
   v = (*VPeak* I / *V Re f* × *cos* (2π *FSignalI* × *t*))
   + *j* (*VPeakQ/V Re f* × *cos* (2π *FSignalQ* × *t*))

$$vs = 2 \times sqrt(Power \times 2 \times ROut) \times v \times e^{j2\pi FCarrier \times t + \pi Phase/180} \times e^{-j2\pi FCarrier \times t}$$

Ramp



Description: Baseband or RF ramp generator
Library: Timed, Sources
Class: TSDF_Ramp
Derived From: basesource
C++ Code: See *doc/sp_items/TSDF_Ramp.html* under your installation directory.

Parameters

Name	Description	Default	Symbol	Unit	Type	Range
ROut	output resistance	DefaultROut		Ohm	real	(0, ∞)
RTemp	physical temperature in degrees C	DefaultRTemp		Celsius	real	[-273.15, ∞)
TStep	simulation time step	0.0001		sec	real	[0, ∞)
FCarrier	carrier frequency	0.0		Hz	real	[0, ∞)
VStart	start voltage level	0.0		V	real	(-∞, ∞)
VFinal	final voltage level	1.0		V	real	(-∞, ∞)
VOff	off voltage level	0.5		V	real	(-∞, ∞)
Type	type of ramp: Linear Ramp, Power Ramp, Exponential Ramp	Linear Ramp			enum	
RampConstant	used for power and exponential type ramp	1			real	(-∞, ∞)
Delay	time delay before turn-on	0.0005		sec	real	[0, ∞)
DurationTime	time duration for ramp-on	0.001	D	sec	real	[2 * TStep, ∞)
RepetitionInterval	repetition time interval	0.002		sec	real	[D, ∞)

Pin Outputs

Pin	Name	Description	Signal Type
1	output	output signal	timed

Notes/Equations

1. For information regarding timed source signals, refer to *Timed Sources* (timed).
2. This source generates an RF (complex envelope) timed signal output when FCarrier > 0, or a baseband timed signal output when FCarrier=0. It is a ramp generator with settable ramp type and repetition interval.
   Let v = basic ramp waveform
   If ROut = 0; then vs = v
   If ROut > 0; then vs = 2×v
3. Ramp voltage types are expressed:
   For time interval $0 \le t \le T_e$

- Linear Ramp
  $V(t) = v\,1 + (v\,2 - v\,1)\ t\ /\ T_e$
- Power Ramp

$$V(t) = v_1 + (v_2 - v_1)\left(\frac{t}{T_e}\right)^k$$

- Exponential Ramp

$$V(t) = v_1 + (v_2 - v_1)\ \frac{1 - e^{-t/k}}{1 - e^{-T_e/k}}$$

For time interval $T_e \le t \le T_i$

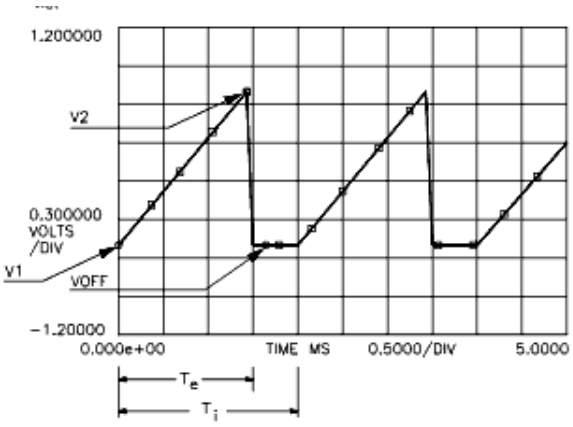
$V\ (t) = v\ 3$
Let
$v\ 1$ = VStart
$v\ 2$ = VFinal
$v\ 3$ = VOff
T_e = DurationTime

T_i = RepetitionInterval

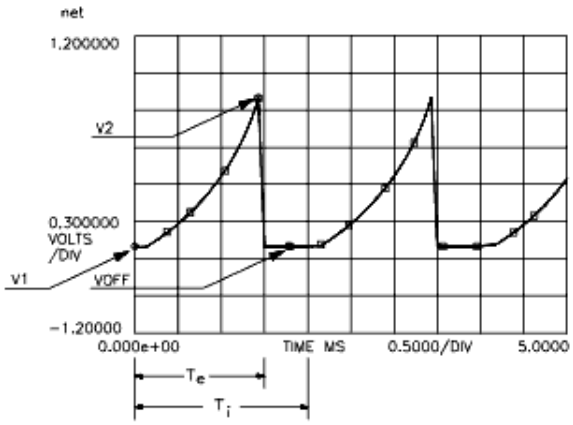
k = RampConstant
t = fmod(time, T_i)

Examples of linear, power, and exponential ramps are shown in Linear Ramp, Power Ramp, and Exponential Ramp, respectively:

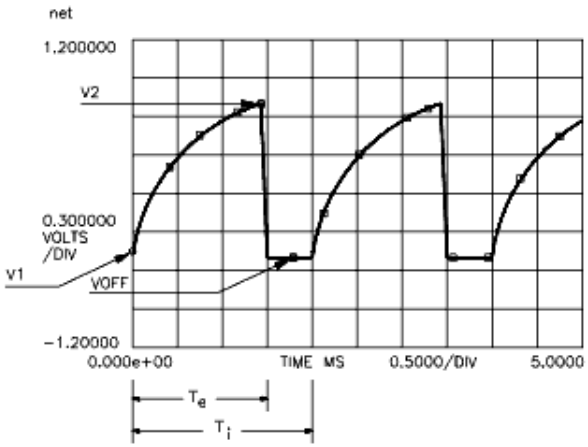
Linear Ramp



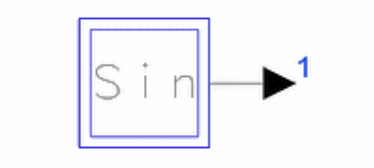
Power Ramp



Exponential Ramp



Sinusoid



Description: Sinusoid signal generator
Library: Timed, Sources
Class: TSDF_Sinusoid
Derived From: basesource
C++ Code: See *doc/sp_items/TSDF_Sinusoid.html* under your installation directory.

Parameters

Name	Description	Default	Symbol	Unit	Type	Range
ROut	output resistance	DefaultROut		Ohm	real	(0, ∞)
RTemp	physical temperature in degrees C	DefaultRTemp		Celsius	real	[-273.15, ∞)
TStep	simulation time step	0.0001		sec	real	[0, ∞)
VPeak	peak voltage level	1.0		V	real	(-∞, ∞)
Frequency	sinusoid frequency	1000.0		Hz	real	[0, ∞)
Phase	phase angle in degrees	0.0		deg	real	(-∞, ∞)
DecayRatio	ratio of decay value at end to beginning of DurationTime, 0 = no decay, 1 = full decay; 0 <= DecayRatio <= 1	0			real	[0, 1]
Delay	turn-on time delay	0.0		sec	real	[0, ∞)
DurationTime	time duration for sinusoid on	0.001	D	sec	real	[2 * TStep, ∞)
RepetitionInterval	repetition time interval	0.002		sec	real	[D, ∞)

Pin Outputs

Pin	Name	Description	Signal Type
1	output	output signal	timed

Notes/Equations

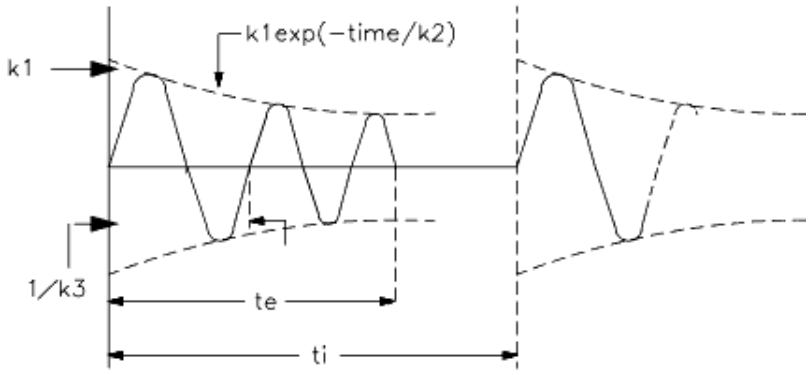
1. For information regarding timed source signals, refer to *Timed Sources* (timed).
2. This source generates a baseband timed signal output. It is a damped sinusoid generator with settable exponential decay.
   Let
   v = basic damped sinusoid waveform
   If ROut = 0, then vs = v
   If ROut > 0, then vs = 2×v
3. SINE Generator illustrates this generator.
   Let
   k1 = VPeak

k3 = Frequency

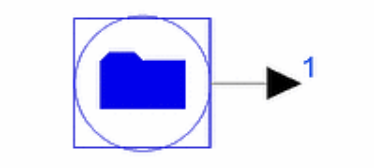
$$\text{k2} = \text{Decay Constant} = -\frac{DurationTime}{\log(1-DecayRatio)}$$

te = DurationTime
ti = RepetitionInterval

SINE Generator



TimedDataRead



Description: Time Domain Signal Generator with File-Based Data
Library: Timed, Sources
Class: TSDFTimedDataRead
Derived From: basesource

Parameters

Name	Description	Default	Unit	Type	Range
TStep	simulation time step (if TStep=0 the time step is calculated using the independent variable values of the first two data samples in the data file)	0.0001	sec	real	[0, ∞)
ControlSimulation	if set to YES, Period (or if Period=0 the timestamp on last data in the file) determines how long the simulation will run: NO, YES	NO		enum	
Periodic	if YES then output is periodic: NO, YES	YES		enum	
Period	period of the output waveform if Periodic=YES. If Period=0 then the entire file is repeated	0	sec	real	[0, ∞)
FileName	input file			filename	
FCarrier	carrier Frequency (for complex data FCarrier must be > 0)	0.0	Hz	real	[0, ∞)

Pin Outputs

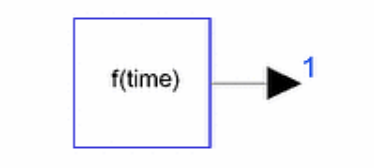
Pin	Name	Description	Signal Type
1	output	output signal	timed

Notes/Equations

1. This source generates a constant amplitude RF (complex envelope) timed signal output when FCarrier > 0, or a baseband timed signal output when FCarrier=0. It is a signal source defined using a time-domain signal file.
2. The *.tim* and *.bintim* files use MDIF format with *.tim* in ASCII form and *.bintim* in binary form. The *.ascsig* and *.sig* files use a signal file format used in the Cadence Signal Processing Workstation product with *.ascsig* in ASCII form and *.sig* in binary form. For format information, refer to *Understanding File Formats* (ptolemy) in the *ADS Ptolemy Simulation* (ptolemy) documentation.
3. Setting the Periodic parameter causes the model to loop back to the beginning of the file. The Period parameter controls how much data is read from the file before it starts looping back to the first data in the file. If Period=0 it will read all data from the file and then repeat it.
4. If ControlSimulation is set to YES and if there is no other source or sink controlling simulation, the simulation may stop when all data in the file has been processed. For more information, refer to *Sources and Sinks Control the Simulation* (ptolemy) in the *ADS Ptolemy Simulation* (ptolemy) documentation.

5. If the TStep of simulation is less than the data in the file being read, interpolation will be performed.
6. The complement of TimedDataRead is *TimedDataWrite* (sinks) (Sinks library), which is used to generate a *.tim*, *.bintim*, *.ascsig*, or *.sig* file.

TimedExpression



Description: Timed Expression Data source
Library: Timed, Sources
Class: TSDFTimedExpression
Derived From: basesource

Parameters

Name	Description	Default	Unit	Type	Range
TStep	simulation time step	0.0001	sec	real	[0, ∞)
FCarrier	carrier frequency	0.0	Hz	real	[0, ∞)
Expression	expression, which can be function of "time"	0.0+j*0.0		complex	

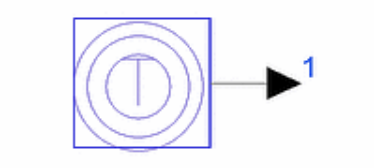
Pin Outputs

Pin	Name	Description	Signal Type
1	output	output signal	timed

Notes/Equations

1. This source is used to generate timed data output that is evaluated using an expression. Expression can be any valid expression, following the syntax used for writing expression on a VAR block. For predefined functions that can be used to build more complicated expressions, refer to the *Simulator Expressions* (expsim) documentation.
   If the Expression is dependent on predefined variable *time*, the output will be dependent on the simulation time, which is incremented by *TStep* for each firing of this component decided by the schedule.
2. Prior to the ADS2002C release, this component allowed non-zero imaginary parts of the Expression parameter even when FCarrier was set to 0; this resulted in generating complex envelope signals with characterization frequency equal to 0. Beginning with the ADS2002C release, when FCarrier is set to 0, the imaginary part of the Expression will be ignored.

TimedSource



Description: Time domain signal generator using dataset
Library: Timed, Sources
Class: TSDFTimedSource
Derived From: TimedDataRead

Parameters

Name	Description	Default	Unit	Type	Range
TStep	simulation time step (if TStep=0 the time step is calculated using the independent variable values of the first two data samples in the data file)	0.0001	sec	real	[0, ∞)
ControlSimulation	if set to YES, Period (or if Period=0 the timestamp on last data in the file) determines how long the simulation will run: NO, YES	NO		enum	
Periodic	if YES then output is periodic: NO, YES	YES		enum	
Period	period of the output waveform if Periodic=YES. If Period=0 then period is the timestamp of last data read	0	sec	real	[0, ∞)
FCarrier	carrier Frequency (for complex data FCarrier must be > 0.)	0.0	Hz	real	[0, ∞)
DataSet	dataSet file to construct Expression from			filename	
Expression	variable/sink name from dataset or a valid DataSet expression (data must be single dimensional)			string	

Pin Outputs

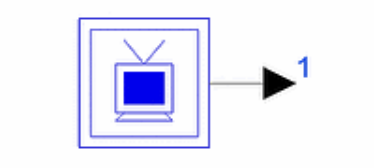
Pin	Name	Description	Signal Type
1	output	output signal	timed

Notes/Equations

1. This source is used to generate timed data output evaluated using an existing dataset. Expression can be any valid expression using variables available in the dataset. The syntax used for writing the expression is the same as writing an expression to display data in a Data Display window. Only variables available in the dataset can be used in the Expression parameter. Variables that are defined through equations in a data display file that has the same dataset as its default dataset cannot be used.
   The expression must evaluate into 1-dimensional data. Any expression that results in higher dimension data will error out. To reduce dimensionality, use the "[...,::,...]" operator.
   For example, consider a design that has a TimedSink *T1* and 3 levels of sweep. If such a dataset is used for generating data using TimedSource and the Expression was set to "T1", the simulation will error out saying it was 4- dimensional data. To fix this error, use "N1[0,0,0,::]", which will now generate 1-dimensional data at the

output.
If TStep does not match the TStep of the data in the dataset, interpolation is performed.
If the length of simulation is larger than the available data in the dataset, use Periodic and Period parameters to repeat the old data. The Periodic parameter must be set to YES for the output to repeat after time equal to Period. If Periodic=YES and Period=0, the Period will be the time stamp on the last data read in the dataset, and all of the data from the dataset will be read and repeated. If Periodic=NO, the output will be zero after all data is read.
If ControlSimulation=YES, Period will determine how long the simulation runs. If Period=0, the simulation will run until the last data in the dataset is read.

Video



Description: Video (NTSC or PAL) signal generator
Library: Timed, Sources
Class: TSDF_Video
Derived From: basesource

Parameters

Name	Description	Default	Unit	Type	Range
ROut	output resistance	DefaultROut	Ohm	real	(0, ∞)
RTemp	physical temperature in degrees C	DefaultRTemp	Celsius	real	[-273.15, ∞)
TStep	simulation time step	0.0001	sec	real	[0, ∞)
Type	type of NTSC or PAL video lines: NTSC 5 Step Linearity, NTSC 10 Step Linearity, NTSC Ramp Linearity, NTSC Sine Squared Pulse and Bar, NTSC Multiburst 90 IRE, NTSC Multiburst 60 IRE, NTSC Composite, NTSC VIRS, NTSC Flat Field 100 IRE, NTSC Flat Field 50 IRE, NTSC Modulated Pedestal, NTSC Color Bar, PAL Line 17 Composite, PAL Line 330 Composite with D1, PAL Line 330 Composite with D2, PAL Multiburst, PAL Chrominance Bar, PAL Modulated Pedestal	NTSC 5 Step Linearity		enum	†
Form	Video line or Video frame: Video line, Video frame	Video line		enum	†
StartField	used when Form=Video frame; use value 1 or 2 (when Type=NTSC Type); use value 1 to 4 (when Type=PAL Type)	1		int	[1, 4]†
StartLine	used when Form=Video frame; use value 1 to 525 (when Type=NTSC type); use value 1 to 625 (when Type=PAL type)	1		int	[1, 625]†

† This parameter must satisfy the conditions
if Form = Video frame
if Type = NTSC type
StartField = 1 or 2
StartLine = 1 to 525
else
StartField = 1, 2, 3, or 4
StartLine = 1 to 625

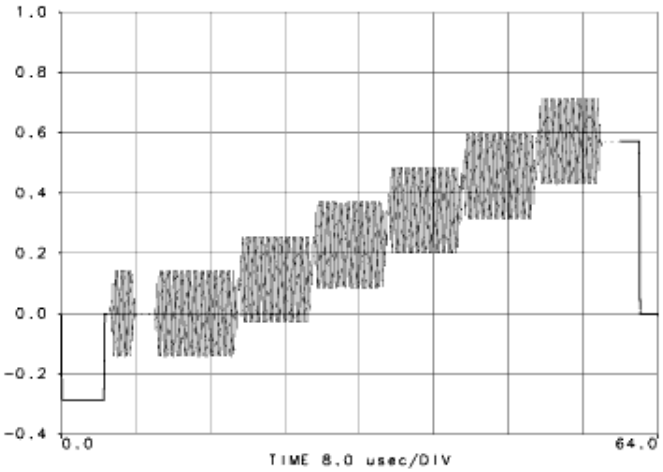
Pin Outputs

Pin	Name	Description	Signal Type
1	output	output signal	timed

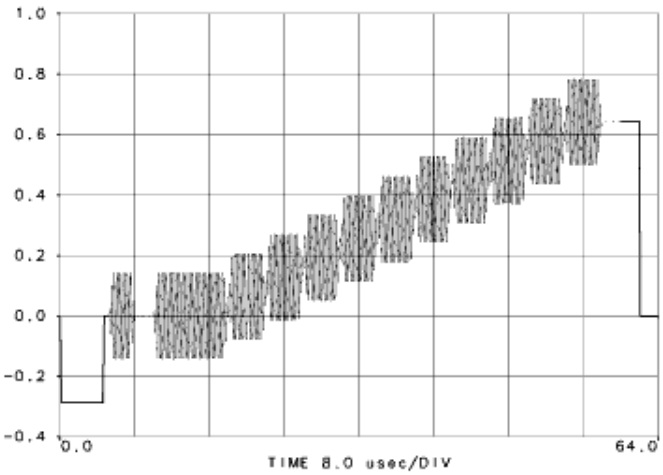
Notes/Equations

1. For information regarding timed source signals, refer to *Timed Sources* (timed).
2. This source generates a baseband timed signal output. It is a video waveform generator for NTSC and PAL type video test signals. EIA standards were used for NTSC; CCIR standards were used for PAL. Video signals can be generated as an individual line or an entire frame.
   Let v = the basic video waveform
   If ROut = 0; then vs = v
   If ROut > 0; then vs = 2 × v
3. NTSC 5-Step Linearity through NTSC Color Bar show the available types of NTSC test signals selected from the Type parameter.
4. PAL Line 17 Composite through PAL Modulated Pedestal show the available types of PAL test signals selected from the Type parameter.

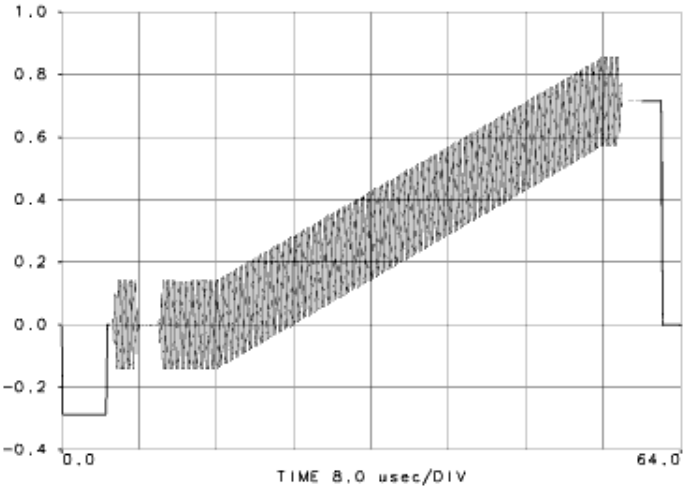
NTSC 5-Step Linearity



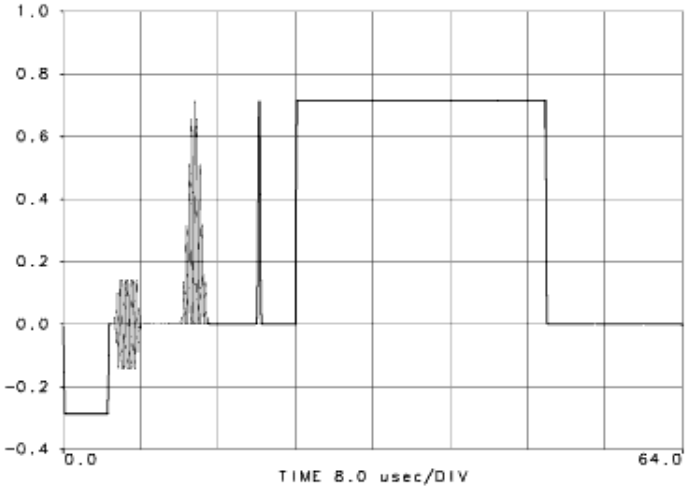
NTSC 10-Step Linearity



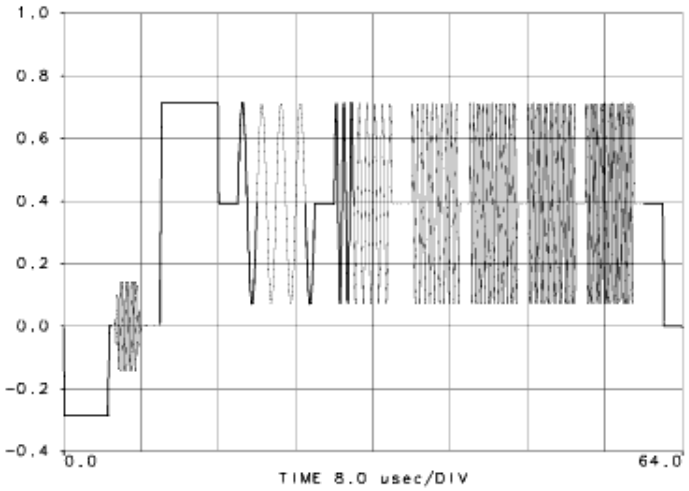
NTSC Ramp Linearity



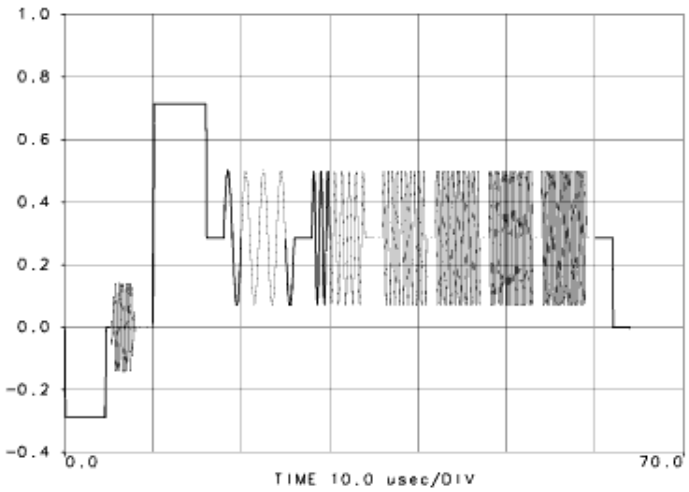
NTSC Sine-Squared Pulse and Bar



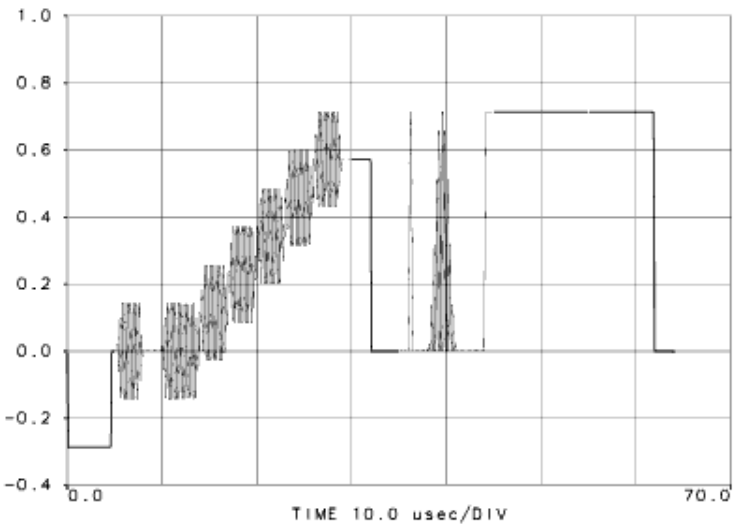
NTSC Multiburst 90 IRE



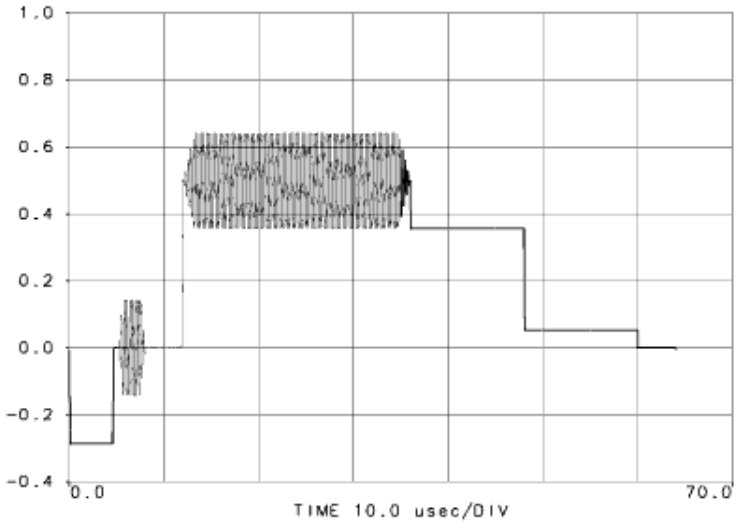
NTSC Multiburst 60 IRE



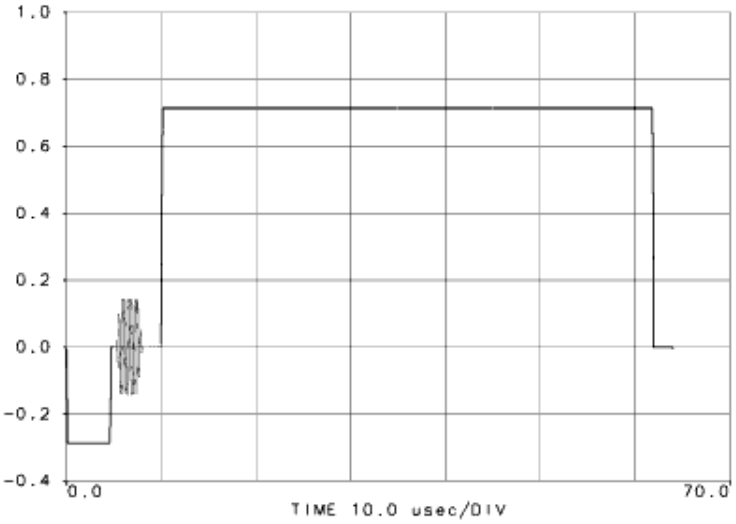
NTSC Composite



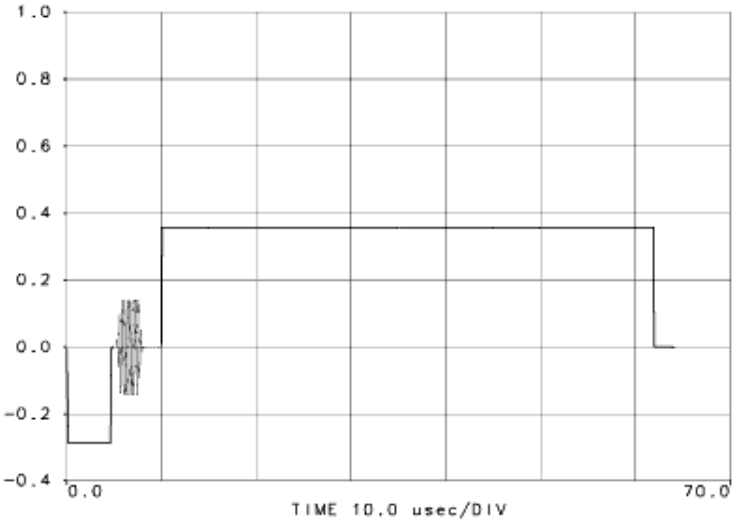
NTSC VIRS



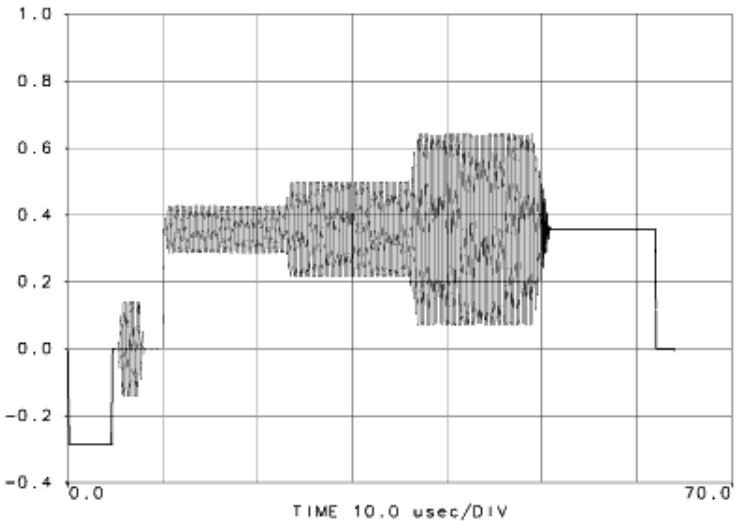
NTSC Flat-Field 100 IRE



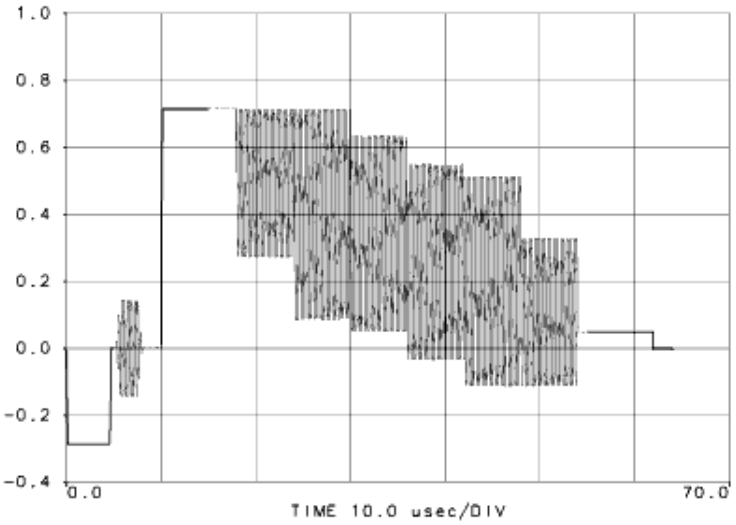
NTSC Flat-Field 50 IRE



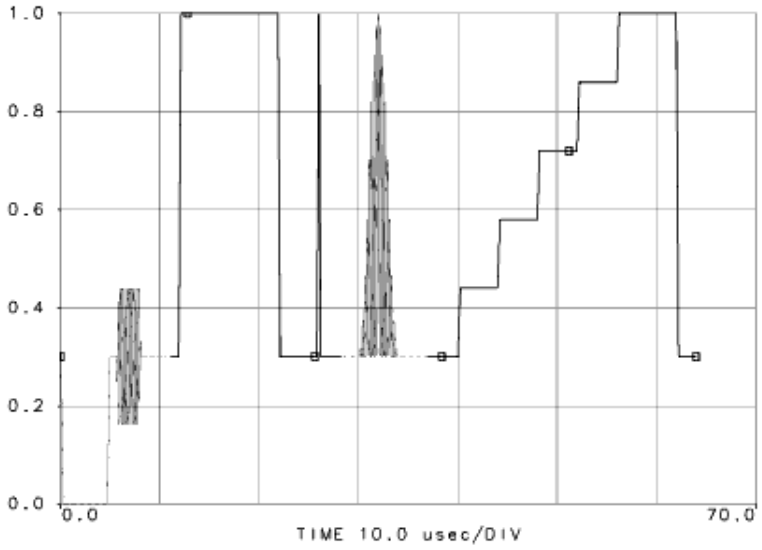
NTSC Modulated Pedestal



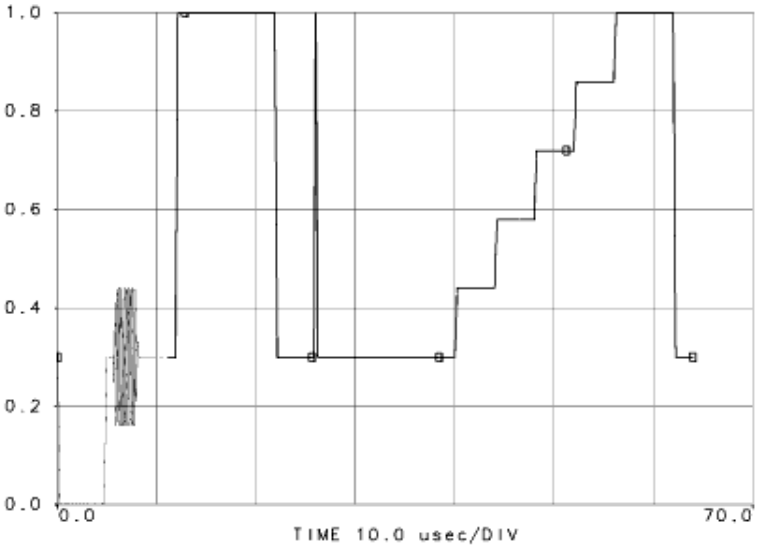
NTSC Color Bar



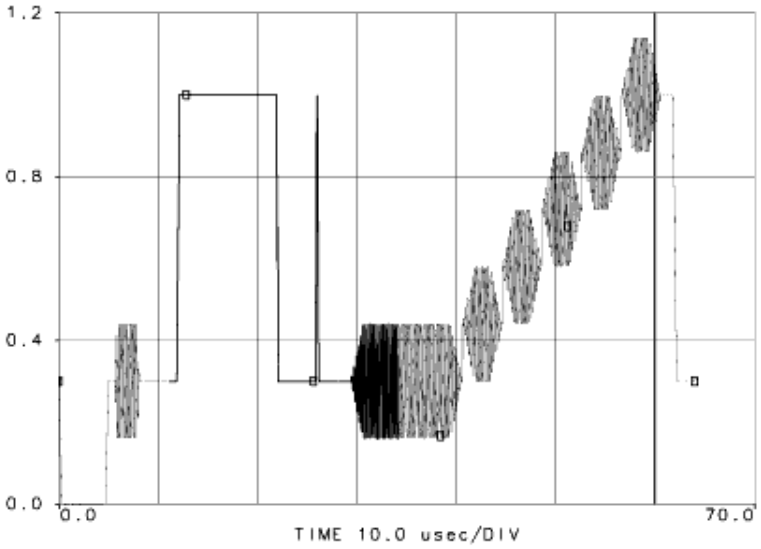
PAL Line 17 Composite



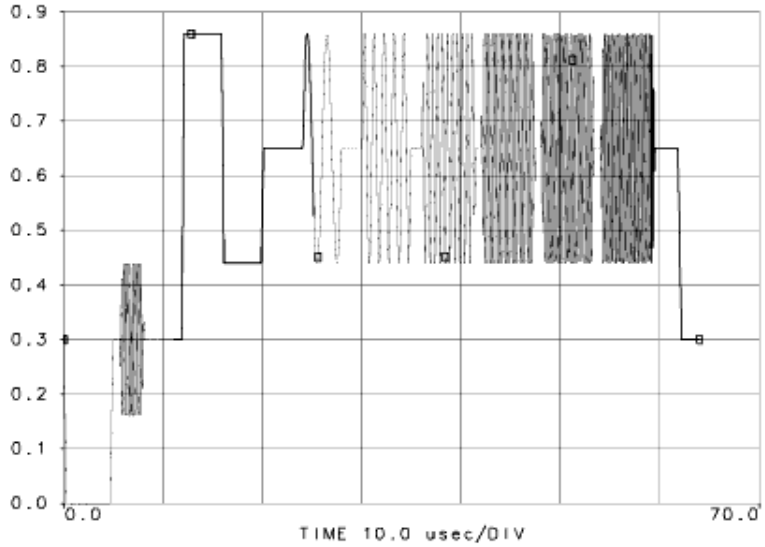
PAL Line 330 Composite with D1



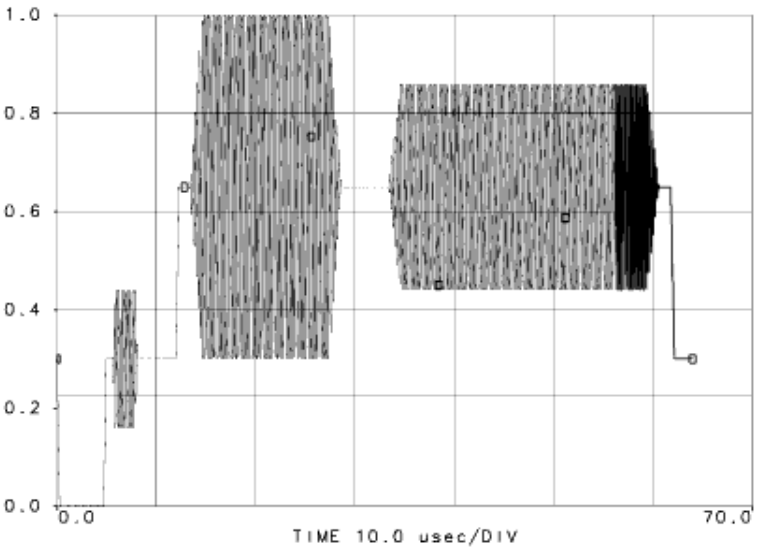
PAL Line 330 Composite with D2



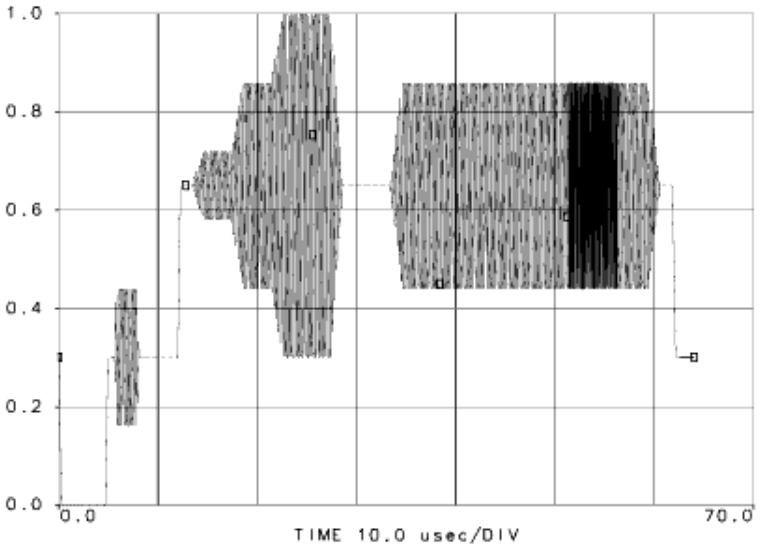
PAL Multiburst



PAL Chrominance Bar



PAL Modulated Pedestal



NTSC References

1. NTC Report No. 7, Video Facility Testing Technical Performance Objectives, prepared by the Network Transmission Committee of the Video Transmission Engineering Advisory Committee, published by the Public Broadcasting Service, January 1976.

PAL References

1. Recommendations of the CCIR, 1990, Volume XI - Part 1, Broadcasting Service (Television), CCIR (International Radio Consultative Committee), Geneva, 1990.
2. Recommendations of the CCIR, 1990, Annex to Volume XI - Part 1, Broadcasting Service (Television), CCIR (International Radio Consultative Committee), Geneva, 1990.
3. Recommendations of the CCIR, 1990, Volume XII, Television and Sound Transmission (CMTT), CCIR (International Radio Consultative Committee), Geneva, 1990.
4. Reports of the CCIR, 1990, Annex to Volume XII, Television and Sound Transmission (CMTT), CCIR (International Radio Consultative Committee), Geneva, 1990.