Twin Builder® Components: Basic Elements VHDLAMS

ANSYS, Inc.
Southpointe
2600 Ansys Drive
Canonsburg, PA 15317
ansysinfo@ansys.com
https://www.ansys.com
(T) 724-746-3304
(F) 724-514-9494

Release 2025 R2
July 2025

ANSYS, Inc. and ANSYS Europe, Ltd. are UL registered ISO 9001:2015 companies.

Table of Contents

Table of Contents **Contents-1**

1 - Basic Elements VHDLAMS Library **1-2**

Load Reference Arrow System of Physical Domains - VHDL-AMS 1-4

Nature Types of Components - VHDL-AMS 1-6

Connecting Components - VHDL-AMS 1-7

Modeling with Block Diagrams 1-8

Connecting Transfer Blocks 1-9

Signal Direction in Block Diagrams 1-10

Defining the Sample Time 1-11

Continuous Blocks 1-12

Delay 1-13

Derivative 1-16

Gain 1-19

S-Transfer Function 1-22

Integrator 1-25

Memory 1-28

Discrete Blocks 1-32

Discrete Derivative 1-33

Discrete Integrator 1-36

Z-Transfer Function (Discrete) 1-39

Sample and Hold Element 1-43

Unit Delay 1-46

Math Blocks 1-49

Absolute Value Function 1-50

Arc Cosine Function 1-53

Cosine Function 1-56

Exponential Function 1-59

Natural Logarithm Function 1-62

Reciprocal Function ... 1-65
Sine Function ... 1-68
Sine Hyperbolic Function ... 1-71
Tangent Function ... 1-74
Power ... 1-77
Root ... 1-80
Sign ... 1-83
Signal Processing Blocks ... 1-86
Comparator ... 1-87
Limiter ... 1-90
Maximum of Input Signals ... 1-93
Minimum of Input Signals ... 1-97
Multiplier ... 1-101
Negator ... 1-105
Summation ... 1-108
Two-point Element with Hysteresis ... 1-112
Source Blocks ... 1-116
Constant Value ... 1-117
Random ... 1-120
Step Function ... 1-123
Modeling with Circuit Components ... 1-126
Load Reference Arrow System of Circuit Components ... 1-127
Network Configurations ... 1-128
Electrical Machines ... 1-129
DC Machine Linear Electrical Excitation ... 1-130
DC Machine Permanent Excitation ... 1-137
Induction Machine with Squirrel Cage Rotor ... 1-144
Synchronous Machine Electrical Excitation without Damper ... 1-151
Synchronous Machine Electrical Excitation with Damper ... 1-158
Synchronous Machine Permanent Excitation without Damper ... 1-164

Synchronous Machine Permanent Excitation with Damper ... 1-169
Passive Elements ... 1-175
Capacitor ... 1-176
Conductor ... 1-181
Inductor ... 1-185
Resistor ... 1-190
Semiconductors System Level ... 1-194
Bipolar Junction Transistor ... 1-195
Diode ... 1-200
IGBT ... 1-204
MOS Fieldeffect Transistor ... 1-208
Sources ... 1-212
Voltage Source ... 1-213
Controlled Voltage Source ... 1-217
Voltage Controlled Oscillator Voltage Source ... 1-221
Current Source ... 1-225
Controlled Current Source ... 1-229
Voltage Controlled Oscillator Current Source ... 1-233
Switches ... 1-237
Current Controlled Switch ... 1-238
Voltage Controlled Switch ... 1-242
Ideal Switch ... 1-246
Ideal Transfer Switch ... 1-250
Transformers ... 1-255
Single-Phase Systems ... 1-257
Primary Side of a Two-Winding Transformer ... 1-258
Secondary Side of a Two-Winding Transformer ... 1-261
Linear Two-Winding Transformer ... 1-264
Single-Phase Transformer Example ... 1-269
Three-Phase Systems ... 1-272

Primary Side of a Six-Winding Transformer ... 1-273
Secondary Side of a Six-Winding Transformer ... 1-278
Six-Winding Transformer Large-Power ... 1-283
Six-Winding Transformer Small-Power ... 1-290
Three-Phase Six-Winding Transformer Example ... 1-297
Using Measuring Instruments ... 1-299
Electrical Meters ... 1-300
Fluidic Meters ... 1-302
Magnetic Meters ... 1-304
Mechanical Meters ... 1-306
Thermal Meters ... 1-311
Modeling with Physical Domains Components - VHDL-AMS ... 1-312
Fluidic Components - VHDL-AMS ... 1-313
Hydraulic Capacitance - VHDL-AMS ... 1-314
Hydraulic Inductance - VHDL-AMS ... 1-317
Pressure Source - VHDL-AMS ... 1-321
Flow Source - VHDL-AMS ... 1-325
Hydraulic Resistance - VHDL-AMS ... 1-329
Magnetic Components - VHDL-AMS ... 1-332
Flux Source - VHDL-AMS ... 1-333
Magneto-Motive Force Source - VHDL-AMS ... 1-336
Magnetoreluctance - VHDL-AMS ... 1-338
Winding - VHDL-AMS ... 1-340
Magnetic Circuit Example - VHDL-AMS ... 1-343
Mechanical Components - VHDL-AMS ... 1-346
Displacement-Force-Representation ... 1-347
Displacement-Force Rotational Components ... 1-348
Damper - VHDL-AMS ... 1-349
Torque Source - VHDL-AMS ... 1-351
Limit Stop - VHDL-AMS ... 1-353

Mass - VHDL-AMS 1-355
Angle Source - VHDL-AMS 1-357
Spacer - VHDL-AMS 1-359
Spring - VHDL-AMS 1-361
Angular Velocity Source - VHDL-AMS 1-363
Angle Source/Spacer/Damper Rotational Displacement-Force Example 1-365
Angular Velocity Source/Spring/Damper Rotational Displacement-Force Example 1-367
Torque Source/Mass/Limit Stop Rotational Displacement-Force Example 1-369
Displacement-Force Translational Components 1-371
Damper - VHDL-AMS 1-372
Force Source - VHDL-AMS 1-374
Limit Stop - VHDL-AMS 1-376
Mass - VHDL-AMS 1-378
Position Source - VHDL-AMS 1-380
AMS Solenoid 1-382
Spacer - VHDL-AMS 1-384
Spring - VHDL-AMS 1-386
Velocity Source - VHDL-AMS 1-388
Position Source/Spacer/Damper Translational Displacement-Force Example 1-390
Velocity Source/Spring/Damper Translational Displacement-Force Example 1-393
Force Source/Mass/Limit Stop Translational Displacement-Force Example 1-395
Velocity-Force-Representation 1-397
Velocity-Force Rotational V Components 1-398
Damper - VHDL-AMS 1-399
Torque Source - VHDL-AMS 1-401
Limit Stop - VHDL-AMS 1-403
Mass - VHDL-AMS 1-405
Spring - VHDL-AMS 1-407
Angular Velocity Source - VHDL-AMS 1-409

Angular Velocity Source/Spring/Damper Rotational Velocity-Force Example 1-411
Torque Source/Mass/Limit Stop Rotational Velocity-Force Example 1-413
Velocity-Force Translational V Components 1-415
Damper - VHDL-AMS 1-416
Force Source - VHDL-AMS 1-418
Limit Stop - VHDL-AMS 1-420
Mass - VHDL-AMS 1-422
Spring - VHDL-AMS 1-424
Velocity Source - VHDL-AMS 1-426
Velocity Source/Spring/Damper Translational Velocity-Force Example 1-428
Force Source/Mass/Limit Stop Translational Velocity-Force Example 1-430
Thermal Components - VHDL-AMS 1-432
Thermal Capacitance - VHDL-AMS 1-433
Heat Flow Source - VHDL-AMS 1-436
Thermal Resistance - VHDL-AMS 1-438
Temperature Source - VHDL-AMS 1-440
Thermal Component Example 1-442
Modeling Tools 1-445
Using Time Functions 1-446
Time Functions 1-447
Arc Tangent 1-448
Needle 1-451
Pulse 1-454
Sawtooth Wave Falling 1-457
Sawtooth Wave Rising 1-460
Sine Wave 1-463
Trapezoidal Wave 1-466
Triangular Wave 1-469
Index **Index-1**

1 - Basic Elements VHDLAMS Library

The Basic Elements VHDLAMS Library has been developed to support the user with the most common basic functionality, circuit components and blocks in VHDL-AMS. The models have been developed according to the IEEE 1076.1 (VHDL Analog and Mixed Signal Extensions Standard).

The functionality of all VHDL-AMS models is a subset of that available with the equivalent SML models. Also, the VHDL-AMS and VHDL models do not use wizards to parameterize the models.

VHDL-AMS models may be used in parallel with Twin Builder models in the **Basic Elements** library. All VHDL-AMS models are simulated by the analog simulator and the digital solver. As a consequence, the VHDL-AMS models will not display step delays along the simulator backplane, unlike the SML models. VHDL-AMS block models as well as the circuit elements are simulated by the analog solver, while Twin Builder blocks are solved by a separate block diagram simulator. As a consequence, the simulation results from both models are not always be identical. The difference in the results usually decreases if the time step of the system, hmin and hmax, are reduced.

Numerical values of the VHDL-AMS data type generic are defined at t = 0 and remain unchanged for the remainder of the simulation.

It is possible to view the inputs and outputs of the VHDL-AMS model using the appropriate analog or digital display element. It is also possible to view the internal values used within the architecture of a model in the display graphs and use them as variables between models. However, if the VHDL-AMS design needs to be exported to other simulators then internal values of the model should not be used outside the model.

The models provided are open code and can be used to derive more advanced models. This may be done by copying the description in an user library and editing the text to modify the model. The files can be used and distributed if the copyright statement, included in each model description, is not removed.

```
-- the fitness for a particular purpose.
--  ----------------------------------------------------------------------
```

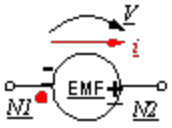
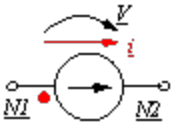
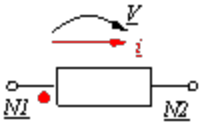
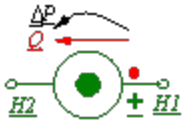
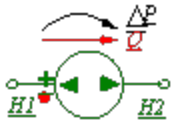
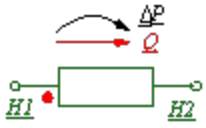
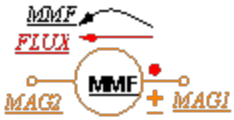
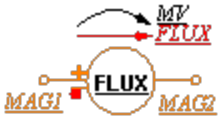
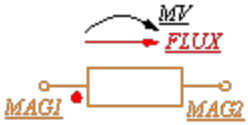
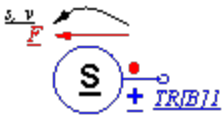
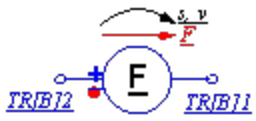
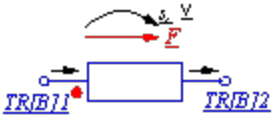
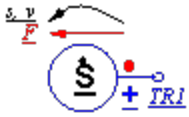
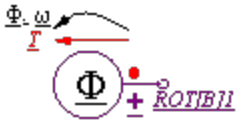
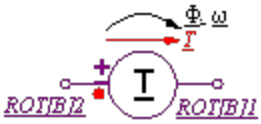
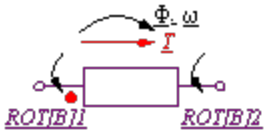
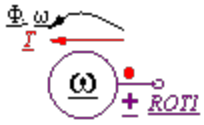
Load Reference Arrow System of Physical Domains - VHDL-AMS

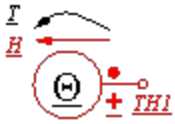
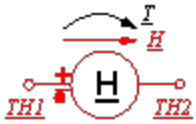
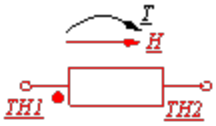
The table shows the reference arrow system for all in Twin Builder available domains.

The counting direction is marked by the red point at the component symbol. The red point is always at pin 1 of a component.

Across quantities: Value=Value at Pin1 – Value at Pin 2.

Through quantities: Value is positive if the quantity flows at the red point into the component.

	Difference Sources	Flow Sources	Passive Components
Electrical			
Fluidic			
Magnetic			
Mechanic Translational			
			
Mechanic Rotational			
			

Thermal			

Nature Types of Components - VHDL-AMS

Nature types are properties from conservative nodes (ports) of components. At least one specific nature exists for each domain. The mechanical domain has four nature types.

In the Schematic, you can connect conservative nodes only if they have the same nature type. By way of contrast, other quantities (non-conservative nodes) can be changed between all domains.

The **Basic Element VHDLAMS** library provides components with the following nature types:

- electrical
- fluidic
- magnetic
- translational
- translational_v
- rotational
- rotational_v
- thermal

Connecting Components - VHDL-AMS

Connections between components are created in the wire mode (CTRL+W or Draw>Wire). Place the cross wire on a component pin on the sheet, and set the beginning, corners, and end of the wire with the mouse. Press ESC to finish the wire mode.

Conservative component nodes must always be connected with a node of the same nature type. To connect conservative nodes of different nature types, you must use the D2D (domain to domain) component.

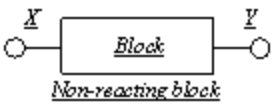
See *General Domain Transformation* in the Transformations Component help.

Schematic tests the pins during the routing and will not allow invalid connections (for example between an electrical and magnetic nature).

After the simulator starts, Schematic tests the complete simulation model. If connections are missing, an error message or warning is displayed in the **Build** area of the Information window. Double-click the warning or error text to move the component with the wrong connection into the visible range of the Schematic.

Modeling with Block Diagrams

Transfer Blocks in Twin Builder are defined by an internal identifier, the name and a set of parameters; Blocks have no conservative nodes. A block is a linear or nonlinear transfer element with a defined static or dynamic response characteristic. The block output signal does not influence the block input quantity (non-reacting).



Block diagrams with typical transfer behaviors can be modeled for control and cybernetic systems by means of the available blocks.

The block output signal and the specific block dependent parameters can be used as outputs. The block output signal is represented by the **VAL** entry; the other parameters by their corresponding name.

For block models, only the input is of type quantity, all other parameters are generics/static values.

The *Blocks* folder provides components in the following categories:

- Continuous Blocks
- Discrete Blocks
- Math Blocks
- Signal Processing Blocks
- Sources

Connecting Transfer Blocks

The structure of the block diagram is created in the Wire mode (CTRL+W or CONNECT>WIRE menu). Schematic tests the pins during the routing and will not allow invalid connections (e.g., between two inputs or two outputs).

If an input or an output in the block diagram is not connected, the **Build** Info Window displays a warning. To avoid this warning, you can hide any unused pins of an element. Open the property dialog and clear the check box of the pin you want to hide, either at the **Parameters** or **Output/Display** page. The pin is invisible on the sheet and the warning no longer occurs.

See also *Using Pins* in the main Twin Builder help.

Signal Direction in Block Diagrams

The sequence of block computation can significantly influence the simulation result. You can change the block sequence with the SHEET>DETERMINE BLOCK SEQUENCE command.

Using Automatic Block Sorting

When the automatic Block sorting mode in SHEET>DETERMINE BLOCK SEQUENCE is active, the blocks will be sorted after simulation start according to their signal direction. If the automatic block sorting is active, the **Build** Info Window displays a message after the simulation start.

Using Manual Block Sorting

To sort blocks manually, clear the option **automatically on start simulation** in SHEET>DETERMINE BLOCK SEQUENCE. Select <Interactive> and the current sequence is displayed in digits next to the blocks. The sequence can be changed by clicking the blocks in the desired sequence on the sheet or by arranging the block list in a new sequence in the dialog.

The following functions are active in the interactive mode:

• Click the block: Defines next position in the sequence.

• Double-click the block: Moves the block to the last position in the list of blocks already sorted.

• Double-click beside the element: Exits the mode and applies the settings.

• <ESC>: Exits the mode without applying the changes.

Note: The sequence of transfer blocks in subsheets must be defined separately. Open the subsheet and start the function SHEET>DETERMINE BLOCK SEQUENCE. The sequence starts in each subsheet with '1'.

Defining the Sample Time

You can choose system sample time ('0') or a specific value for each block. A proper sample time is recommended for giving a consistent behavior for the block diagrams during the simulation.The sample time for the VHDL-AMS blocks are static values (generics). Consequently, this parameter value cannot be expressed as a variable or expression and it cannot be exposed as a pin.

Note: The sampling time must be greater than the minimum time step (HMIN) and less than the simulation end time (TEND). If the sampling time is defined with **System**, the blocks are calculated with the time step determined by the other modules (electric circuit, state graph, or equations), but at least by the maximum time step of integration.

Continuous Blocks

- Delay Block in VHDL-AMS (delay)
- Derivative Block in VHDL-AMS (diff)
- Gain Block in VHDL-AMS (gain)
- S-Transfer Function Block in VHDL-AMS (gs)
- Integrator Block in VHDL-AMS (intg)
- Memory Block in VHDL-AMS (memory)

Delay

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Parameters

• Input/Output Quantities

• Example

• References

Description

The block represents a delay between the input and output values. The delay is specified as simulation time. If td=0, the input is transferred to the output without any delay.

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

COUPL delay ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (y0 := @y0 , td := @td , input := @input) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:-:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value[Unit]
td	Delay Time	real	1.0e-3 [s]
y0	Initial Value	real	0.0

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Value	Output	real
input	Input Signal	Input	real

Top

Example

This example illustrates the output of Continuous Block Delay Function

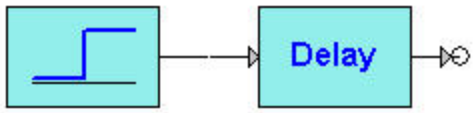


Figure 2. Application example of the VHDL-AMS Delay Function

Table 3. System Parameters

Component	Parameter	Value [unit]
Delay Function delay1	y0	0
	td	2ms
	input	step1.val
Step Function step1	T0	0 [S]

	ampl	1

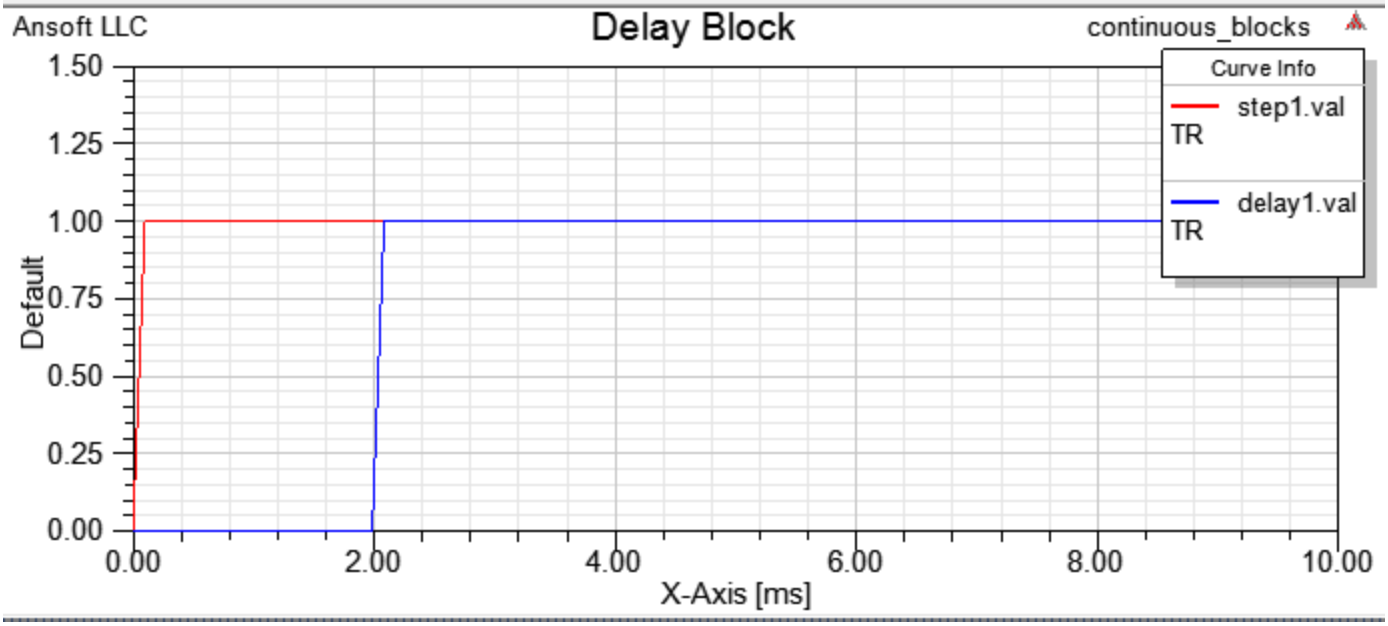


Figure 3. Simulation results-output from function block.

Top

References

Derivative

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Parameters

• Input/Output Quantities

• Example

• References

Description

The block represents a differentiator that is specified with a differential gain. To define the derivative gain, enter a numerical value, a variable, or expression in the parameter list. The initial value represents the output value at t=0. The sample time is a static value. Enter a numerical value in the parameter list. If '0' is specified, the simulator-defined variable sample time is used. The sample time must be greater than HMIN.

Top

Assumptions and Limitations

Top

Mathematical Description

The Laplace-domain transfer function of this component is:

G(s) = kd × s, where kd is the Derivative Gain specified by users.

or we can express the relation between the input and output signal as:

y(k) = kd ×/ ts × [x(n)-x(n-1)], where y(n) and x(n) are the input and output signal at simulation time step n, respectively, and ts is the sampling time.

Top

Netlist Syntax

COUPL diff ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (y0 := @y0 , ts := @ts , input := @input , kd := @kd) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value[Unit]
kd	Differential Gain	real	1.0
y0	Initial Value for t=0	real	0.0
ts	Sample Time	real	0.0 [s]

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Block output signal	Output	real
input	Input Signal	Input	real

Top

Example

This example illustrates the output of Continuous Block Derivative Function

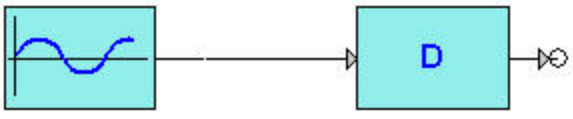


Figure 2. Application example of the VHDL-AMS Derivative Function

Table 3. System Parameters

Component	Parameter	Value [unit]
Derivative Function diff2	kd	1
	ts	0
	input	sine2.val
Sine Function sine2	freq	2 [Hz]
	ampl	1

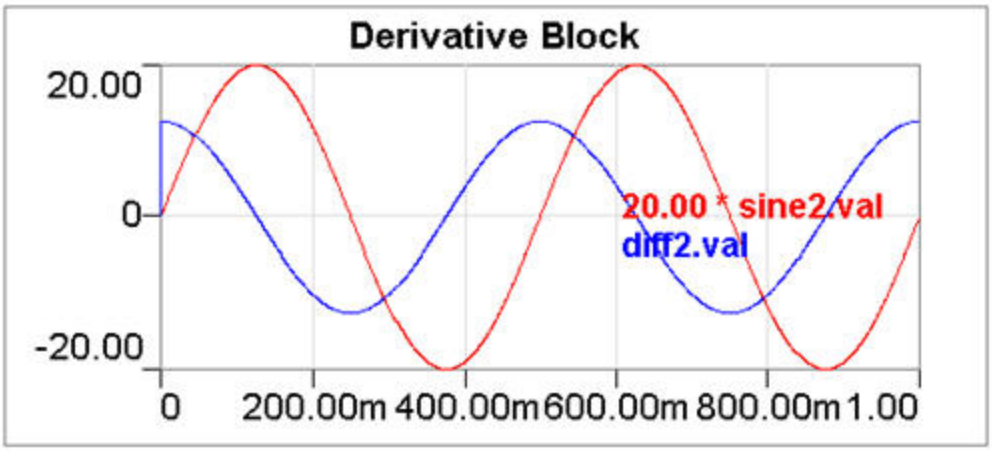


Figure 3. Simulation results-output from function block.

Top

References

Gain

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Parameters

• Input/Output Quantities

• Example

• References

Description

The block represents a proportional gain. To define the gain value, enter a numerical value, a variable, or expression in the parameter list. The sample time is a static value. Enter a numerical value in the parameter list. If '0' is specified, the simulator-defined variable sample time is used. The sample time must be greater than HMIN.

Top

Assumptions and Limitations

Top

Mathematical Description

The Laplace-domain transfer function of this component is:

G(s) = k, where k is the Proportional Gain specified by users.

or we can express the relation between the input and output signal as:

y(n) = k × x(n), where y(n) and x(n) are the input and output signal at simulation time step n, respectively.

Top

Netlist Syntax

COUPL gain ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (ts := @ts , input := @input , kp := @kp) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value[Unit]
k	Proportional Gain	real	1.0
ts	Sample Time	real	0.0 [s]

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Block output signal	Output	real
input	Input Signal	Input	real

Top

Example

This example illustrates the output of Continuous Block Gain Function

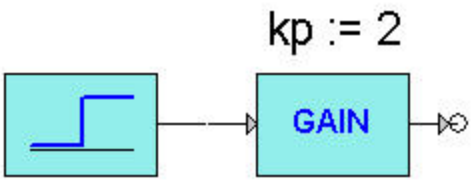


Figure 2. Application example of the VHDL-AMS Gain Function

Table 3. System Parameters

Component	Parameter	Value [unit]

Gain Function gain1	kp	2
	ts	0
	input	step1.val
Step Function step1	T0	0 [S]
	ampl	1

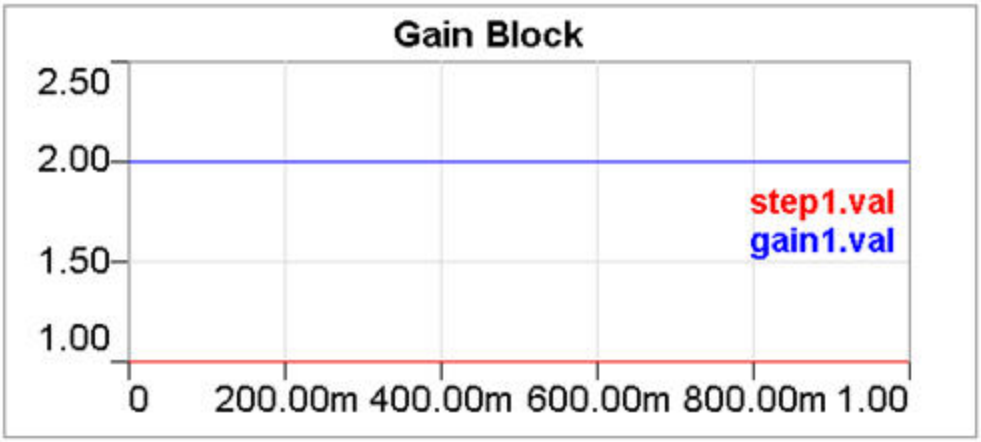


Figure 3. Simulation results-output from function block.

Top

References

S-Transfer Function

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Parameters

• Input/Output Quantities

• Example

• References

Description

The block represents a continuous transfer characteristic G(s). The block uses the LTF attribute to compute the Laplace Transform of the input value. Within the dialog you can define numerator and denominator order and coefficients of the S-Transfer Function.

Note: In order to check LTF function usages for gs under Definitions/Models in the Project Manager, you must first run an analysis.

The sample time is a static value. Enter a numerical value in the parameter list. If '0' is specified, the simulator-defined variable sample time is used. The sample time must be greater than HMIN.

Top

Assumptions and Limitations

Top

Mathematical Description

The Laplace-domain transfer function of this component can be expressed as:

$$G(s) = \frac{num[0] + num[1] \cdot s + num[2] \cdot s^2 + \ldots + num[n] \cdot s^n}{den[0] + den[1] \cdot s + den[2] \cdot s^2 + \ldots + den[m] \cdot s^m} \quad \text{Where} \quad den[m] \neq 0$$

Top

Netlist Syntax

Netlist generated by Special Component Dialog.

Top

Parameters

Table 1

Name	Description	Data Type	Default Value [Unit]
n	Numerator Dimension	real	1
d	Denominator Dimension	real	2
num	Numerator Coefficients	real vector	--
num[%i]	Numerator Coefficients (slice)	real	0.0
den	Denominator Coefficients	real vector	--
den[%i]	Denominator Coefficients (slice)	real	0.0
ts	Sample Time	real	0.0 [s]

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Value	Output	real
input	Input Signal	Input	real

Top

Example

This example illustrates the output of Continuous Block S-Transfer Function

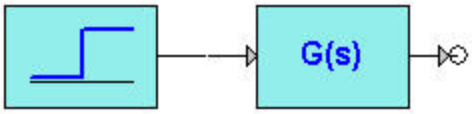


Figure 2. Application example of the VHDL-AMS S-Transfer Function

Table 3. System Parameters

Component	Parameter	Value [unit]
S-Transfer Function gs1	n	0
	d	1
	input	step1.val
Step Function step1	T0	0 [S]
	ampl	1

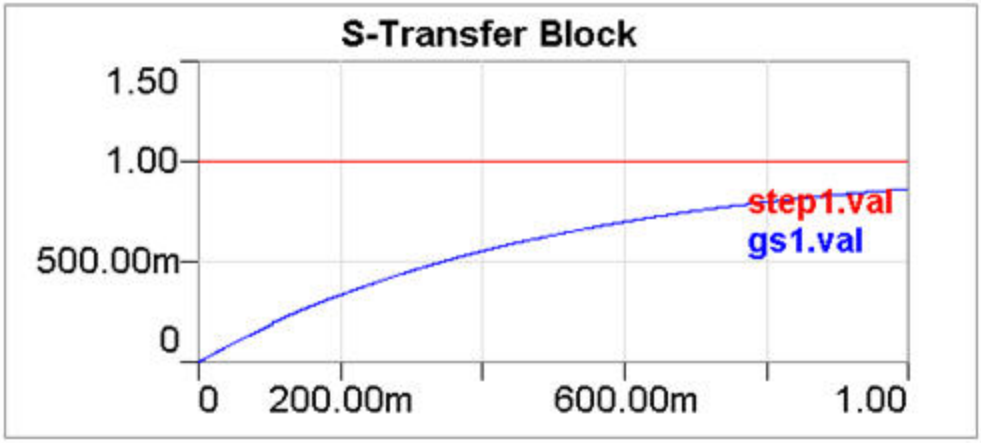


Figure 3. Simulation results-output from function block.

Top

References

Integrator

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Parameters

• Input/Output Quantities

• Example

• References

Description

The block represents an integrator that is specified with an integral gain.

To define the integral gain, enter a numerical value, a variable, or expression in the parameter list. The initial value represents the output value at t=0.

The sample time is a static value. Enter a numerical value in the parameter list. If '0' is specified, the simulator-defined variable sample time is used. The sample time must be greater than HMIN.

Top

Assumptions and Limitations

The Laplace-domain transfer function of this component is:

G(s) = ki/s, where ki is the Integral Gain specified by users.

or we can express the relation between the input and output signal as:

y(n) = y(n-1) + ki × ts × x(n), where y(n) and x(n) are the input and output signal at simulation time step n, respectively, and ts is the sampling time.

Top

Mathematical Description

Top

Netlist Syntax

COUPL intg ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (y0 := @y0 , ts := @ts , input := @input , ki := @ki) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value[Unit]
ki	Integral Gain	real	1.0
y0	Initial Value for t=0	real	0.0
ts	Sample Time	real	0.0 [s]

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Block output signal	Output	real
input	Input Signal	Input	real

Top

Example

This example illustrates the output of Continuous Block Integrator Function

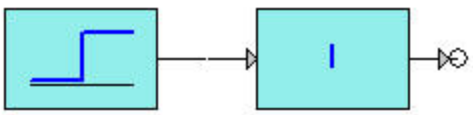


Figure 2. Application example of the VHDL-AMS Integrator Function

Table 3. System Parameters

Component	Parameter	Value [unit]
Integrator Function intg1		1
	ts	0
	input	step1.val
Step Function step1	T0	0 [S]
	ampl	1

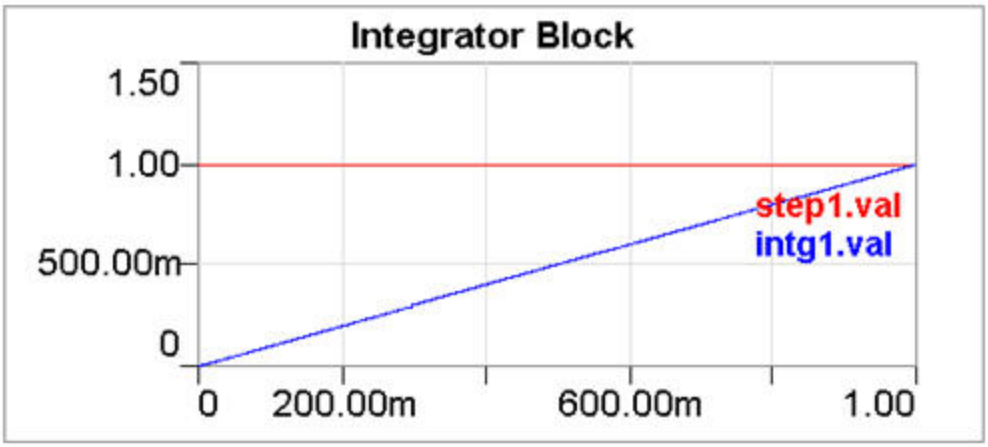


Figure 3. Simulation results-output from function block.

Top

References

Memory

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Parameters

• Input/Output Quantities

• Example

• References

Description

The block represents a delay of one simulation step (H) of the input signal with initial value. The initial value represents the output value at t=0. You cannot define a special sample time for the block.

The memory model has two architectures:

• The "behav" architecture is a pure VHDL-AMS description and always uses the hmin value for delaying the input signal.

• The "sml_behav" architecture is similar to the Simulink memory model and uses a foreign model in the VHDL-AMS description that delays the input with h, the dynamic simulation step size. This architecture can be used to simply break an algebraic loop and delay a signal for exactly one time step, independent of its size.

For exporting pure VHDL-AMS models, the "behav" architecture has to be used to maintain model exchangeability across simulators.

Top

Assumptions and Limitations

Top

Mathematical Description

We can express the relation between the input and output signal as:

y(k) = x(k-1), where y(k) is the output signal at simulation time step k, and x(k-1) is the input signal at the previous simulation time step k-1.

Top

Netlist Syntax

COUPL memory ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (y0 := @y0 , ts := @ts , input := @input) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:-:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value[Unit]
y0	Initial Value for t=0	real	0.0
ts	Sample Time	real	0.0 [s]

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Value	Output	real
input	Input Signal	Input	real

Top

Example

This example illustrates the output of Continuous Block Memory Function

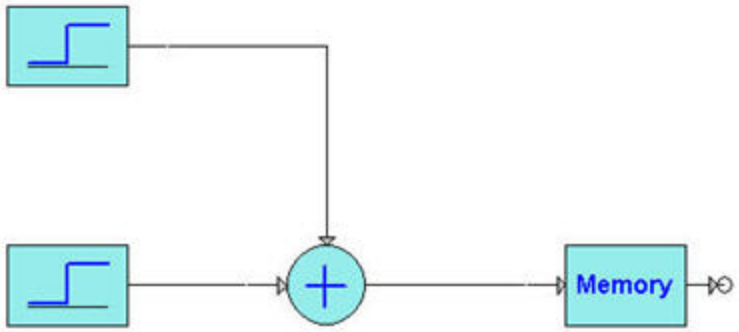


Figure 2. Application example of the VHDL-AMS Memory Function

Table 3. System Parameters

Component	Parameter	Value [unit]
Memory Function memory1	y0	5
	ts	h
	input	sum1.val
Step Function step2	t0	1m [S]
	ampl	1
Step Function step3	t0	4m [S]
	ampl	-1
SUM sum1	input[0]	step3.val
	input[1]	step2.val

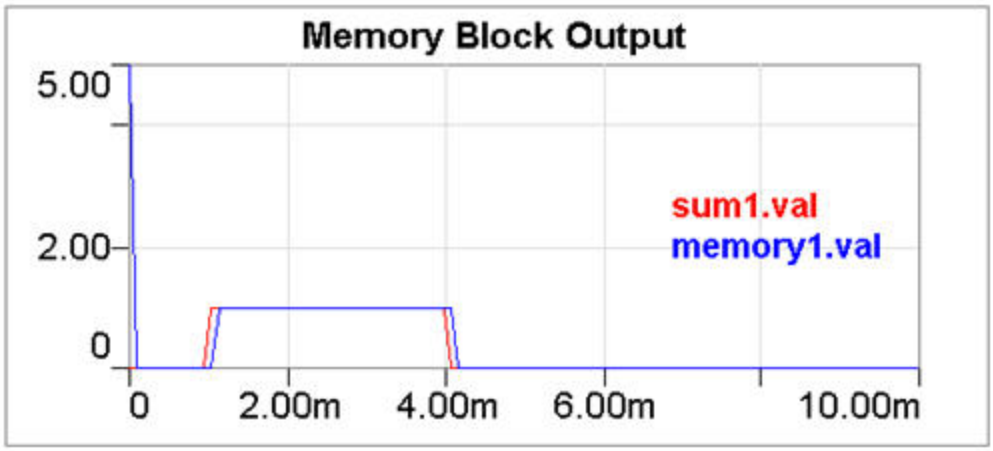


Figure 3. Simulation results-output from function block.

Top

References

Discrete Blocks

- Derivative Block in VHDL-AMS (ddiff)
- Discrete Integrator Block in VHDL-AMS (dintg)
- Z-Transfer Function Block in VHDL-AMS (gz)
- Sample & Hold Block in VHDL-AMS (sah)
- Unit Delay Block in VHDL-AMS (udelay)

Discrete Derivative

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Parameters

• Input/Output Quantities

• Example

• References

Description

The block represents a differentiator that is specified with a differential gain. The block performs the differentiation operation in the conventional method of taking the difference in the input signal and dividing it by the elapsed time.

To define the derivative gain, enter a numerical value, a variable, or expression in the parameter list. The initial value represents the output value at t=0.

The sample time is a static value. Enter a numerical value in the parameter list. The sample time cannot be specified as '0'. If a '0' value is specified, the model provides an error message. The sample time must be greater than HMIN.

The discrete derivative model is different from the continuous derivative in that it does not use the VHDL-AMS statement for differentiation. It computes the differentiated value by sampling the input at two different times and computing the difference in input over time. Consequently, a time value that is not equal to zero has to be specified.

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

COUPL ddiff ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (y0 := @y0 , ts := @ts , kd := @kd , input := @input) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value[Unit]
kd	Derivative Gain	real	1.0
y0	Initial Value for t=0	real	0.0
ts	Sample Time	real	1.0e-3 [s]

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Value	Output	real
input	Input Signal	Input	real

Top

Example

This example illustrates the output of Discrete Block Derivative Function

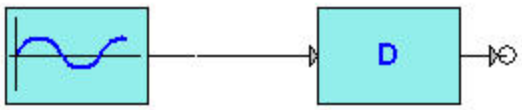


Figure 2. Application example of the VHDL-AMS Derivative Function

Table 3. System Parameters

Component	Parameter	Value [unit]
Discrete Derivative Function ddiff1	kd	1
	ts	1m [S]
	input	sine2.val
Sine Function sine2	freq	50 [Hz]
	ampl	1

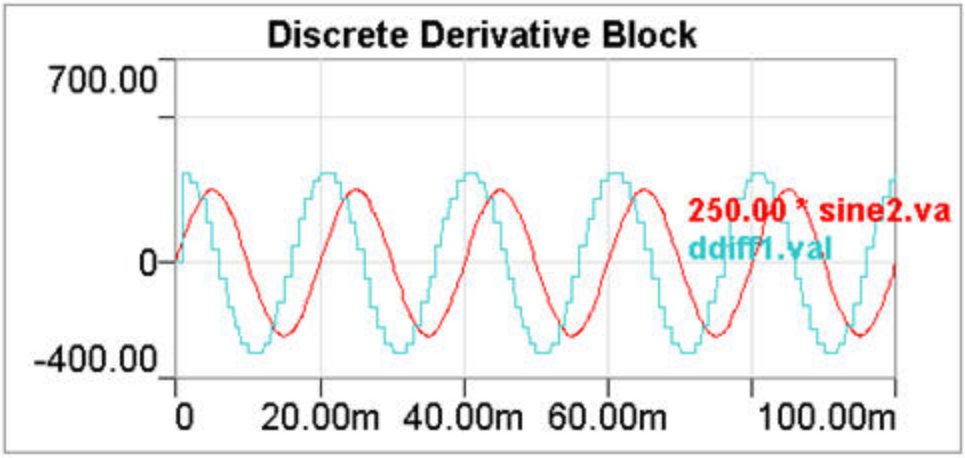


Figure 3. Simulation results-input and output of the Discrete Derivative block.

Top

References

Discrete Integrator

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Parameters

• Input/Output Quantities

• Example

• References

Description

The block represents an integrator that is specified with an integral gain. To define the integral gain, enter a numerical value, a variable, or expression in the parameter list. The initial value represents the output value at t=0.

The sample time is a static value. Enter a numerical value in the parameter list. The sample time cannot be specified as '0'. If a '0' value is specified, the model provides an error message. The sample time must be greater than HMIN.

The discrete integrator model is different from the continuous integrator in that it does not use the VHDLAMS statement for differentiation/integration. It computes the integrated value by sampling the input at two different times and computing the integral through trapezoidal rule. Consequently, a time value that is not equal to zero has to be specified.

Top

Assumptions and Limitations

Top

Mathematical Description

We can express the relation between the input and output signal as:

$$y(n) = y(n-1) + ki \cdot ts \cdot x(n)$$

where x(n) and y(n) are the input and output signal at simulation time step n, respectively. ki is the Integral Gain, and ts is the Sample Time.

Top

Netlist Syntax

COUPL dintg ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (y0 := @y0 , ts := @ts , ki := @ki , input := @input) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value[Unit]
ki	Integral Gain	real	1.0
y0	Initial Value for Output	real	0.0
ts	Sample Time	real	1.0e-3 [s]

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Value	Output	real
input	Input Signal	Input	real

Top

Example

This example illustrates the output of Discrete Block Integrator Function

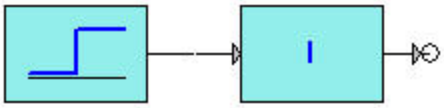


Figure 2. Application example of the VHDL-AMS Integrator Function

Table 3. System Parameters

Component	Parameter	Value [unit]
Discrete Integrator Function dintg1	ki	1
	ts	1m [S]
	input	step1.val
Step Function step1	t0	20m [S]
	ampl	1

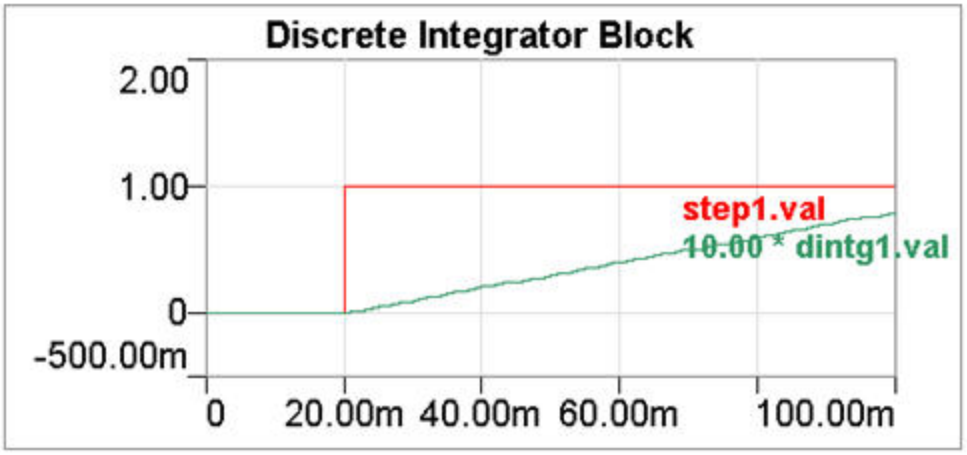


Figure 3. Simulation results-output from function block.

Top

References

Z-Transfer Function (Discrete)

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Parameters

• Input/Output Quantities

• Example

• References

Description

The block represents a time discrete transfer characteristic G*(z). The block uses the ZTF attribute to compute the Z-Domain Transform of the input value.

Note: In order to check ZTF function usages for gz under Definitions/Models in the Project Manager, you must first run an analysis.

Within the dialog you can define numerator and denominator order and coefficients of the Z-Transfer Function.

The sample time is a static value. Enter a numerical value in the parameter list. If '0' is specified, the simulator-defined variable sample time is used. The sample time must be greater than HMIN.

Top

Assumptions and Limitations

Top

Mathematical Description

The transfer function of this component is:

$$G(z^{-1}) = \frac{num[0] + num[1] \cdot z^{-1} + num[2] \cdot z^{-2} + \ldots + num[n] \cdot z^{-n}}{den[0] + den[1] \cdot z^{-1} + den[2] \cdot z^{-2} + \ldots + den[m] \cdot z^{-m}}$$

or we can express the relation between the input and the output as:

$$den[0] \cdot y(k) = num[0] \cdot x(k) + num[2] \cdot x(k-1) + \ldots + num[n] \cdot x(k-n) + den[1] \cdot y(k-1) + den[2] \cdot y(k-2) + \ldots + den[m] \cdot y(k-m)$$

where x(k) and y(k) are the input and output signal at simulation time step k, respectively.

Top

Netlist Syntax

Netlist generated by Special Component Dialog.

Top

Parameters

Table 1

Name	Description	Data Type	Default Value [Unit]
n	Numerator Dimension	integer	1
d	Denominator Dimension	integer	1
num	Numerator Coefficients	real vector	--
num[%i]	Numerator Coefficients (slice)	real	0.0
den	Denominator Coefficients	real vector	--
den[%i]	Denominator Coefficients (slice)	real	0.0
td	Initial Delay	real	0.0
ts	Sample Time	real	0.0 [s]

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Value	Output	real
input	Input Signal	Input	real

Top

Example

This example illustrates the output of Discrete Block Z-Transfer Function

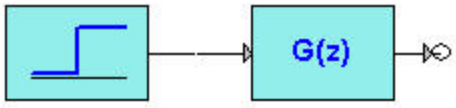


Figure 2. Application example of the VHDL-AMS Z-Transfer Function

Table 3. System Parameters

Component	Parameter	Value [unit]
Z-Transfer Function gz1	n	2
	d	1
	num[0]	1
	num[1]	1
	den[0]	1
	input	step1.val
Step Function step1	T0	0 [S]
	ampl	1

Figure 3. Simulation results-output from function block.

Top

References

Sample and Hold Element

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Parameters

• Input/Output Quantities

• Example

• References

Description

The block samples the input signal and holds this value constant up to the next sampling moment (discretization by time). The sampling period is specified as a number of samples.

To define the sampling period, enter a numerical value in the parameter list. The value must be a positive integer. The initial value represents the output value at t=0.

The sample time is a static value. Enter a numerical value in the parameter list. If '0' is specified, the simulator-defined variable sample time is used. The sample time must be greater than HMIN.

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

COUPL sah ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (y0 := @y0 , ts := @ts , td := @td , input := @input) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value[Unit]
td	Delay Time	real	0.0
y0	Initial Value for t=0	real	0.0
ts	Sample Time	real	0.0 [s]

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Value	Output	real
input	Input Signal	Input	real

Top

Example

This example illustrates the output of Discrete Block Sample and Hold Function

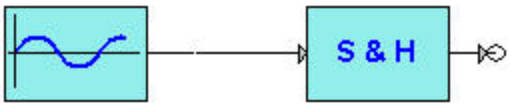


Figure 2. Application example of the VHDL-AMS Discrete Sample and Hold Function

Table 3. System Parameters

Component	Parameter	Value [unit]
Sample and Hold Function sah1	y0	2

	ts	10m [S]
	td	20m [S]
	input	sine1.val
Sine Function sine1	freq	10 [Hz]
	ampl	5

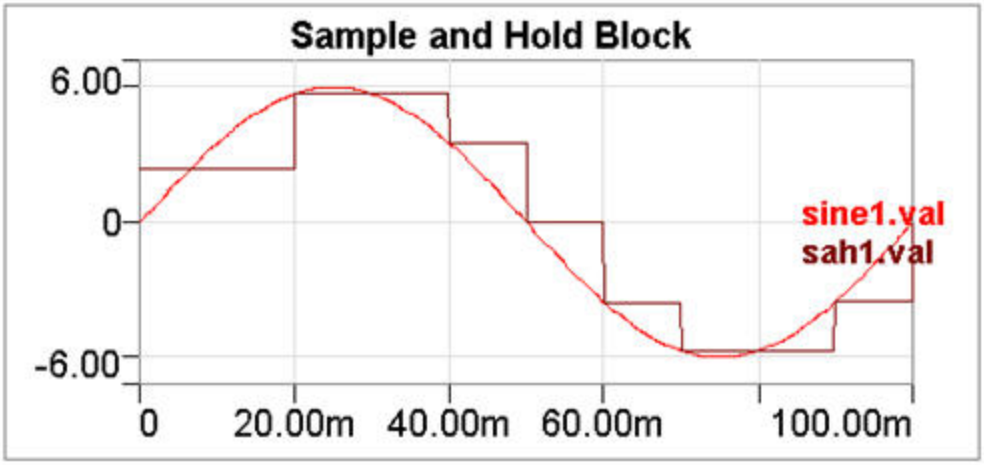


Figure 3. Simulation results-input and output of the Sample and Hold block.

Top

References

Unit Delay

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Parameters

• Input/Output Quantities

• Example

• References

Description

The block represents a delay of the input signal with the as sample time defined time. You can define an initial value for the output signal.

The sample time is a static value. Enter a numerical value in the parameter list. The sample time cannot be specified as '0'. If '0' is specified, the input is transferred to the output without any delay. The sample time must be greater than HMIN.

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

COUPL udelay ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (ts := @ts , y0 := @y0 , input := @input) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName,

Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value[Unit]
y0	Initial Value for t=0	real	0.0
ts	Sample Time	real	0.0 [s]

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Value	Output	real
input	Input Signal	Input	real

Top

Example

This example illustrates the output of Discrete Block Unit Delay Function

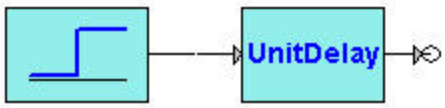


Figure 2. Application example of the VHDL-AMS Unit Delay Function

Table 3. System Parameters

Component	Parameter	Value [unit]
Unit Delay Function udelay1	ts	1m [S]
	input	step1.val
Step Function step1	t0	20m [S]

	ampl	1

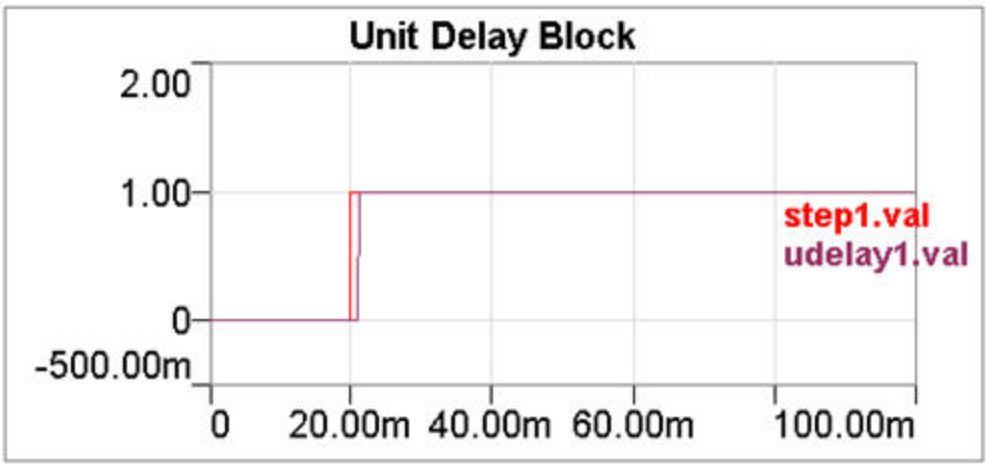


Figure 3. Simulation results-input and output of the Unit Delay block.

Top

References

Math Blocks

- Absolute Value Block in VHDL-AMS (fktabs)
- Arc Cosine Block in VHDL-AMS (fktarccos)
- Cosine Block in VHDL-AMS (fktcos)
- Exponential Function Block in VHDL-AMS (fktexp)
- Natural Logarithm Block in VHDL-AMS (fktln)
- Reciprocal Block in VHDL-AMS (fktres)
- Sine Block in VHDL-AMS (fktsine)
- Sine Hyperbolic Block in VHDL-AMS (fktsinh)
- Tangent Block in VHDL-AMS (fkttan)
- Power Block in VHDL-AMS (pow)
- Root Block in VHDL-AMS (rootn)
- Sign Block in VHDL-AMS (sign)

Absolute Value Function

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Parameters

• Input/Output Quantities

• Example

• References

Description

This block returns the absolute value of the input signal.

The sample time is a static value. Enter a numerical value in the parameter list. If '0' is specified, the simulator-defined variable sample time is used. The sample time must be greater than HMIN.

Top

Assumptions and Limitations

Top

Mathematical Description

The output of the Absolute Value Function block can be expressed as following:

$$y(k) = abs\left(x(k)\right)$$

where y(k) and x(k) are the output and input signal at the simulation time step k, respectively.

Top

Netlist Syntax

COUPL fktabs ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (ts := @ts , input := @input) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:-:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value[Unit]
ts	Sample Time	real	0.0

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Value	Output	real
input	Input Signal	Input	real

Top

Example

This example illustrates the output of Absolute Value Function Math Block.

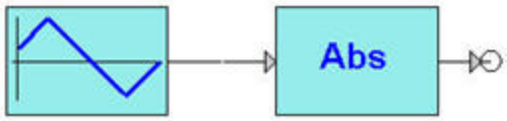


Figure 2. Application example of the VHDL-AMS Absolute Value Math Block

Table 3. System Parameters

Component	Parameter	Value [unit]
Absolute Value Function Math Block fktabs1	ts	0 [S]
	input	triang1.val
Triangular Wave Block triang1	freq	1 [Hz]
	ampl	1
	tdelay	0

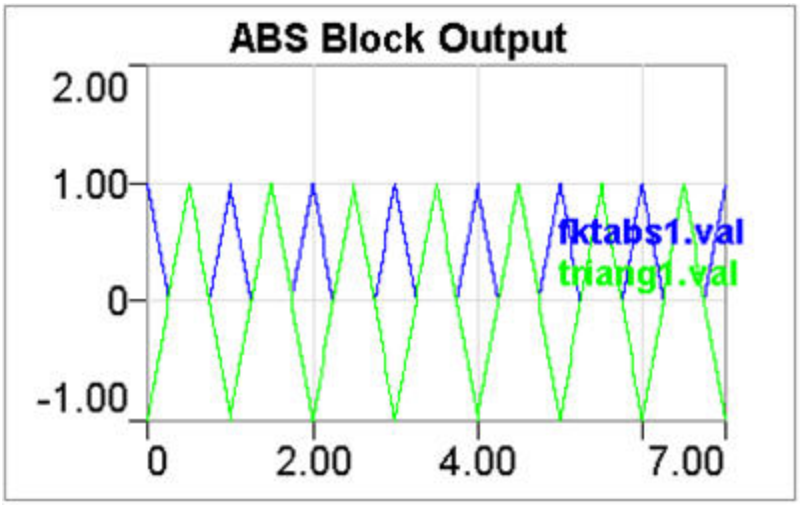


Figure 3. Simulation results-output from the Absolute Value Function block.

Top

References

Arc Cosine Function

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Parameters

• Input/Output Quantities

• Example

• References

Description

This block calculates the standard arc cosine trigonometric function of the input signal.

The sample time is a static value. Enter a numerical value in the parameter list. If '0' is specified, the simulator-defined variable sample time is used. The sample time must be greater than HMIN.

Top

Assumptions and Limitations

Top

Mathematical Description

The output of the Arc Cosine Function block can be expressed as following:

$$y(k) = \arccos\left(x(k)\right)$$

where y(k) and x(k) are the output and input signal at the simulation time step k, respectively.

Top

Netlist Syntax

COUPL fktarccos ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (ts := @ts , input := @input) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:-:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value[Unit]
ts	Sample Time	real	0.0

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Value	Output	real
input	Input Signal	Input	real

Top

Example

This example illustrates the output of Arc Cosine Function Math Block.

Figure 2. Application example of the VHDL-AMS Arc Cosine Math Block

Table 3. System Parameters

Component	Parameter	Value [unit]
Arc Cosine Function Math Block fktar-ccos1	ts	0 [S]
	input	lmt1.val
Gain Block gain2	ts	0
	input	(t/2)-1
	kp	1
Limiter Block lmt1	ts	0
	input	gain2.val
	ul	0.9999
	ll	-1

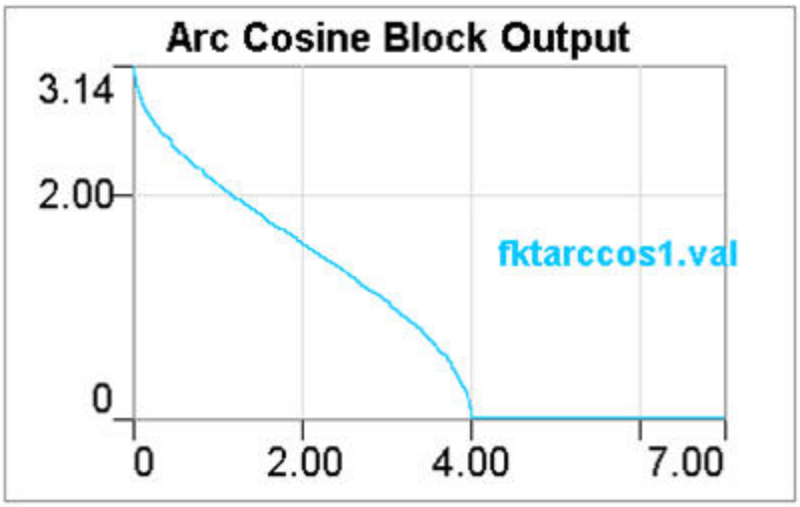


Figure 3. Simulation results-output from Arc Cosine Function block.

Top

References

Cosine Function

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Parameters

• Input/Output Quantities

• Example

• References

Description

This block calculates the standard cosine trigonometric function of the input signal.

The sample time is a static value. Enter a numerical value in the parameter list. If '0' is specified, the simulator-defined variable sample time is used. The sample time must be greater than HMIN.

Top

Assumptions and Limitations

Top

Mathematical Description

The output of the Cosine Function block can be expressed as following:

$$y(k) = \cos\left(x(k)\right)$$

where y(k) and x(k) are the output and input signal at the simulation time step k, respectively.

Top

Netlist Syntax

COUPL fktcos ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (ts := @ts , input := @input) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:-:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value[Unit]
ts	Sample Time	real	0.0

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Value	Output	real
input	Input Signal	Input	real

Top

Example

This example illustrates the output of Cosine Function Math Block.

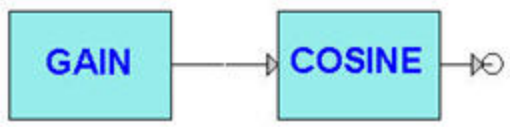


Figure 2. Application example of the VHDL-AMS Cosine Math Block

Table 3. System Parameters

Component	Parameter	Value [unit]
Cosine Function Math Block fktcos1	ts	0 [S]
	input	gain1.val

Gain Block gain1	ts	0
	input	t
	kp	1

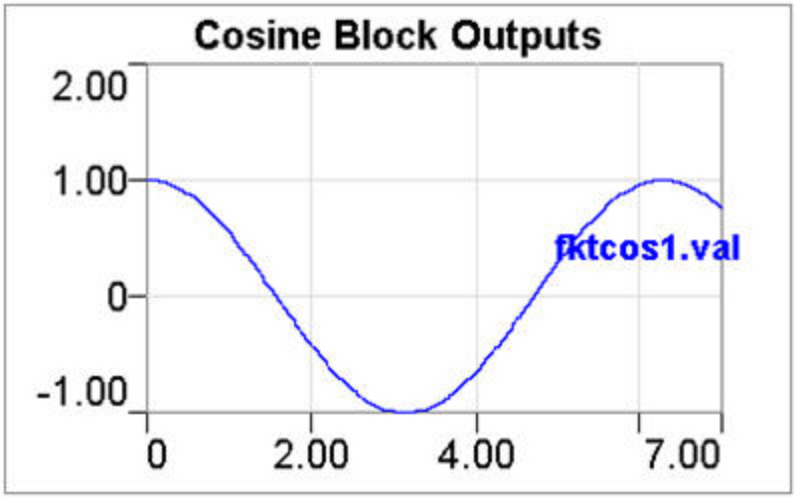


Figure 3. Simulation results-output from Cosine Function block.

Top

References

Exponential Function

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Parameters

• Input/Output Quantities

• Example

• References

Description

The function returns the value corresponding to the exponential of the input at each time step.

The sample time is a static value. Enter a numerical value in the parameter list. If '0' is specified, the simulator-defined variable sample time is used. The sample time must be greater than HMIN.

Top

Assumptions and Limitations

Top

Mathematical Description

The output of the Exponential Function block can be expressed as following:

$$y(k) = \exp\left(x(k)\right)$$

where y(k) and x(k) are the output and input signal at the simulation time step k, respectively.

Top

Netlist Syntax

COUPL fktexp ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (ts := @ts , input := @input) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:-:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value[Unit]
ts	Sample Time	real	0.0

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Value	Output	real
input	Input Signal	Input	real

Top

Example

This example illustrates the output of Exponential Function Math Block.

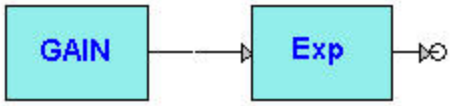


Figure 2. Application example of the VHDL-AMS Exponential Math Block

Table 3. System Parameters

Component	Parameter	Value [unit]
Exponential Function Math Block fkt-exp1	ts	0 [S]
	input	gain2.val
Gain Block gain2	ts	0
	input	(t/2)-1
	kp	1

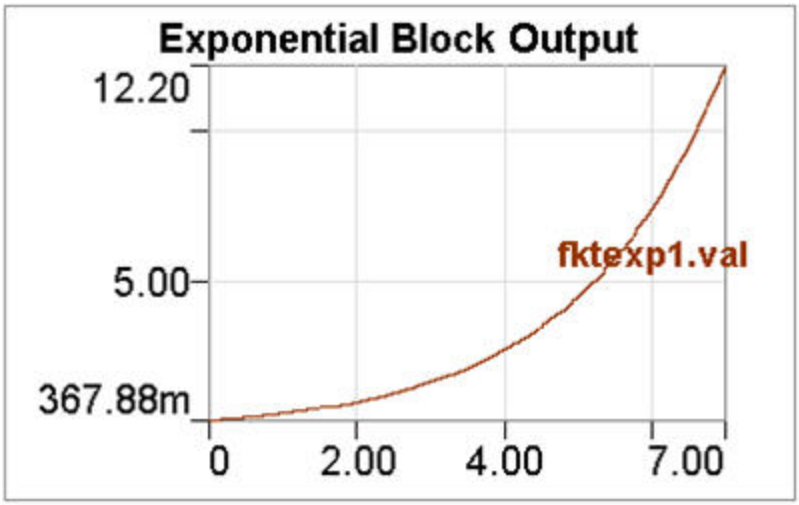


Figure 3. Simulation results-output from Exponential Function block.

Top

References

Natural Logarithm Function

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Parameters

• Input/Output Quantities

• Example

• References

Description

This function returns the natural logarithm of the input signal.

The sample time is a static value. Enter a numerical value in the parameter list. If '0' is specified, the simulator-defined variable sample time is used. The sample time must be greater than HMIN.

Top

Assumptions and Limitations

Top

Mathematical Description

The output of the Natural Logarithm Function block can be expressed as following:

$$y(k) = \ln\left(x(k)\right)$$

where y(k) and x(k) are the output and input signal at the simulation time step k, respectively.

Top

Netlist Syntax

COUPL fktln ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (ts := @ts , input := @input) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:-:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value[Unit]
ts	Sample Time	real	0.0

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Value	Output	real
input	Input Signal	Input	real

Top

Example

This example illustrates the output of Natural Logarithm Function Math Block.

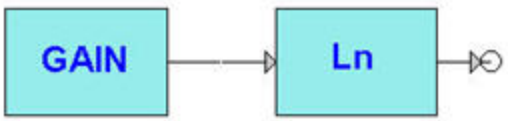


Figure 2. Application example of the VHDL-AMS Natural Logarithm Math Block

Table 3. System Parameters

Component	Parameter	Value [unit]
Natural Logarithm Function Math Block fktln1	ts	0 [S]
	input	gain3.val
Gain Block gain3	ts	0
	input	t+0.1
	kp	1

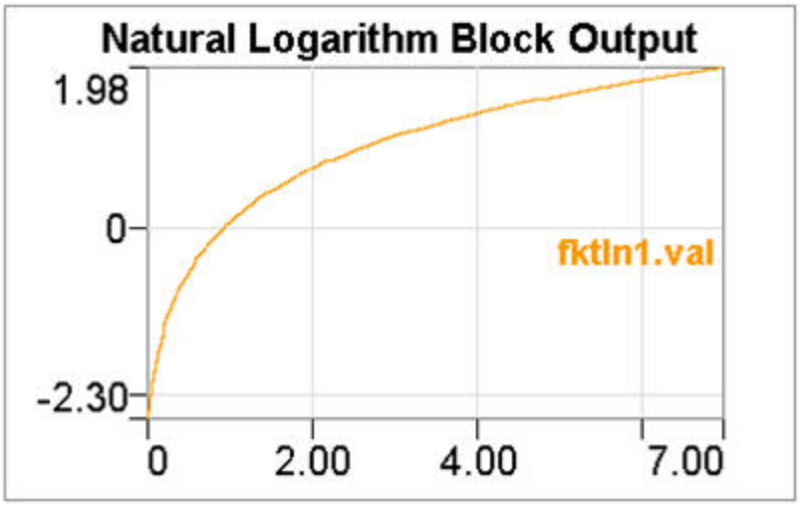


Figure 3. Simulation results-output from the Natural Logarithm block.

Top

References

Reciprocal Function

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Parameters

• Input/Output Quantities

• Example

• References

Description

This block calculates the Reciprocal value (1/X) of the input signal.

The sample time is a static value. Enter a numerical value in the parameter list. If '0' is specified, the simulator-defined variable sample time is used. The sample time must be greater than HMIN.

Top

Assumptions and Limitations

Top

Mathematical Description

The output of the Reciprocal Function block can be expressed as following:

$$y(k) = 1/\left(x(k)\right)$$

where y(k) and x(k) are the output and input signal at the simulation time step k, respectively.

Top

Netlist Syntax

COUPL fktres ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (ts := @ts , input := @input) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:-:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value[Unit]
ts	Sample Time	real	0.0

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Value	Output	real
input	Input Signal	Input	real

Top

Example

This example illustrates the output of Reciprocal Function Math Block.

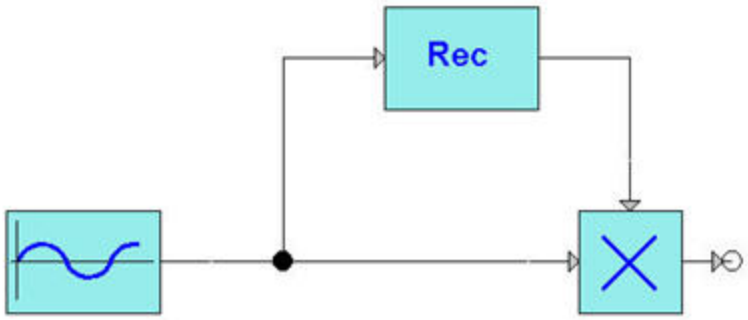


Figure 2. Application example of the VHDL-AMS Reciprocal Math Block

Table 3. System Parameters

Component	Parameter	Value [unit]

Reciprocal Function Math Block fktres1	ts	0 [S]
	input	sine1.val
Sine Block sine1	off	2
	ampl	1
	freq	1 [Hz]
Multiplier Block mul1	ts	0
	input0	sine1.val
	input1	fktres1.val

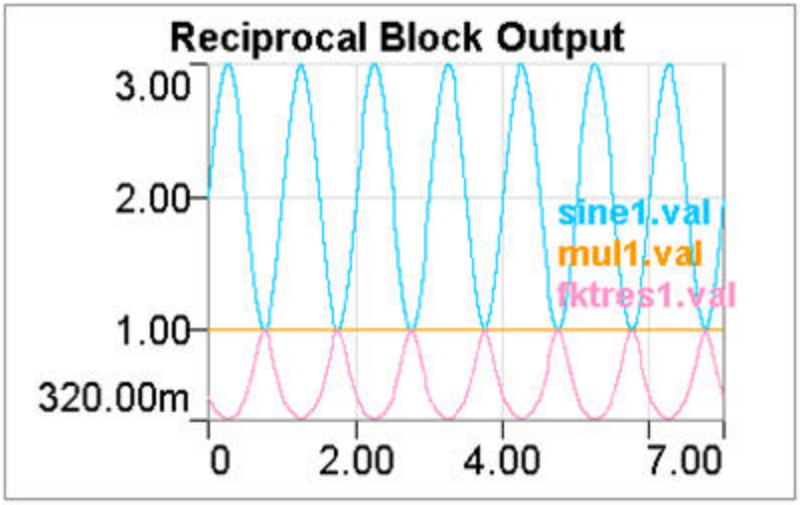


Figure 3. Simulation results-output from the Reciprocal Function block.

Top

References

Sine Function

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Parameters

• Input/Output Quantities

• Example

• References

Description

This block calculates the standard sine trigonometric function of the input signal.

The sample time is a static value. Enter a numerical value in the parameter list. If '0' is specified, the simulator-defined variable sample time is used. The sample time must be greater than HMIN.

Top

Assumptions and Limitations

Top

Mathematical Description

The output of the Sine Function block can be expressed as following:

$$y(k) = \sin(x(k))$$

where y(k) and x(k) are the output and input signal at the simulation time step k, respectively.

Top

Netlist Syntax

COUPL fktsine ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (ts := @ts , input := @input) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:-:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value[Unit]
ts	Sample Time	real	0.0

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Value	Output	real
input	Input Signal	Input	real

Top

Example

This example illustrates the output of Sine Function Math Block.

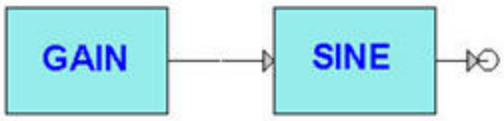


Figure 2. Application example of the VHDL-AMS Sine Math Block

Table 3. System Parameters

Component	Parameter	Value [unit]
Sine Function Math Block fktsine1	ts	0 [S]
	input	gain1.val
Gain Block gain1	ts	0
	input	t
	kp	1

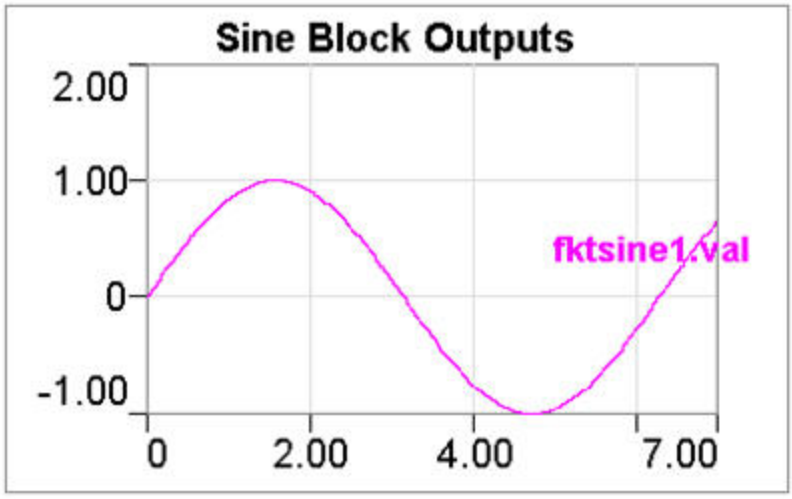


Figure 3. Simulation results-output from the Sine Function block.

Top

References

Sine Hyperbolic Function

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Parameters

• Input/Output Quantities

• Example

• References

Description

This block calculates the standard hyperbolic sine trigonometric function of the input signal.

The sample time is a static value. Enter a numerical value in the parameter list. If '0' is specified, the simulator-defined variable sample time is used. The sample time must be greater than HMIN.

Top

Assumptions and Limitations

Top

Mathematical Description

The output of the Sine Hyperbolic Function block can be expressed as following:

$$y(k) = \sinh\left(x(k)\right)$$

where y(k) and x(k) are the output and input signal at the simulation time step k, respectively.

Top

Netlist Syntax

COUPL fktsinh ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (ts := @ts , input := @input) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:-:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value[Unit]
ts	Sample Time	real	0.0

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Value	Output	real
input	Input Signal	Input	real

Top

Example

This example illustrates the output of Hyperbolic Sine Function Math Block.

Figure 2. Application example of the VHDL-AMS Hyperbolic Sine Math Block

Table 3. System Parameters

Component	Parameter	Value [unit]
Sine Hyperbolic Function Math Block fktsinh1	ts	0 [S]
	input	gain2.val

Gain Block gain2	ts	0
	input	(t/2)-1
	kp	1

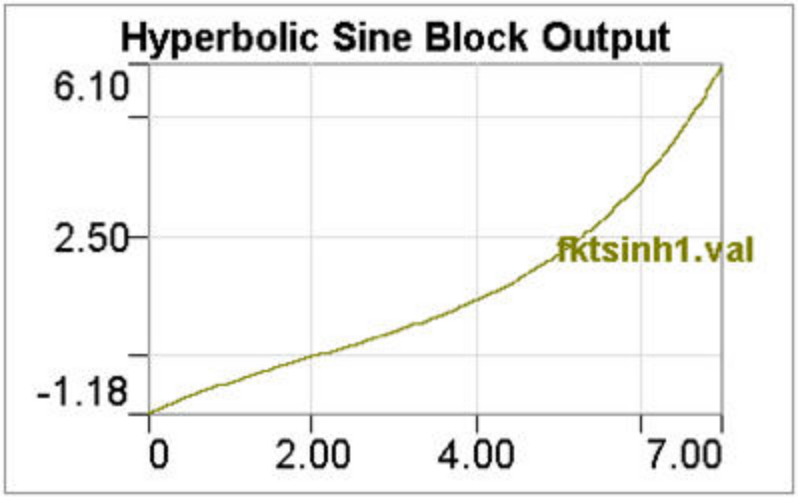


Figure 3. Simulation results-output from Hyperbolic Sine block.

Top

References

Tangent Function

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Parameters

• Input/Output Quantities

• Example

• References

Description

This block calculates the standard tangent trigonometric function of the input signal.

The sample time is a static value. Enter a numerical value in the parameter list. If '0' is specified, the simulator-defined variable sample time is used. The sample time must be greater than HMIN.

Top

Assumptions and Limitations

Top

Mathematical Description

The output of the Tangent Function block can be expressed as following:

$$y(k) = tg(x(k))$$

where y(k) and x(k) are the output and input signal at the simulation time step k, respectively.

Top

Netlist Syntax

COUPL fkttan ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (ts := @ts , input := @input) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:-:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value[Unit]
ts	Sample Time	real	0.0

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Value	Output	real
input	Input Signal	Input	real

Top

Example

This example illustrates the output of Tangent Function Math Block.

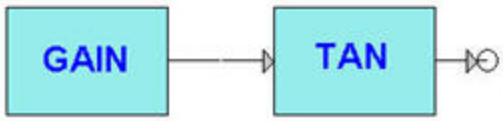


Figure 2. Application example of the VHDL-AMS Tangent Math Block

Table 3. System Parameters

Component	Parameter	Value [unit]
Tangent Function Math Block fkttan1	ts	0 [S]
	input	gain1.val
Gain Block gain1	ts	0
	input	t
	kp	1

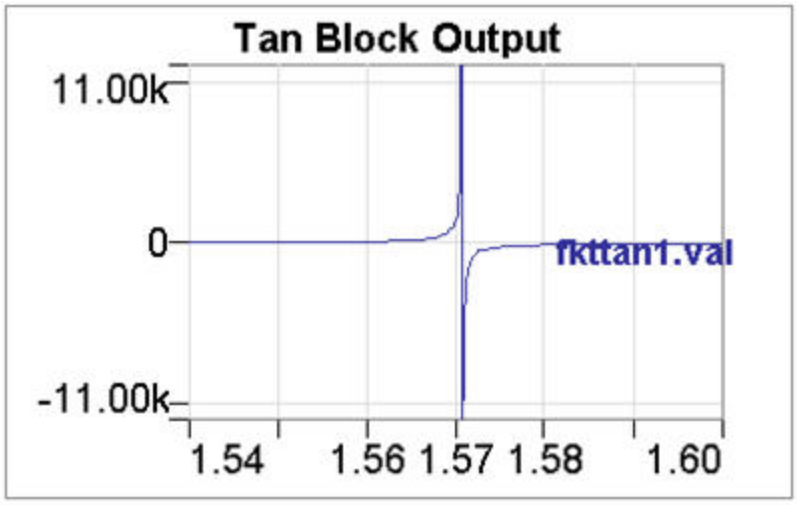


Figure 3. Simulation results-output from Tangent Function block.

Top

References

Power

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

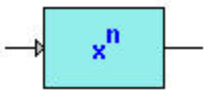


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Parameters

• Input/Output Quantities

• Example

• References

Description

The block raises the block input signal to the n-th power.

To define the exponent value enter a numerical value, a variable, or expression in the parameter list.

The sample time is a static value. Enter a numerical value in the parameter list. If '0' is specified, the simulator-defined variable sample time is used. The sample time must be greater than HMIN.

Top

Assumptions and Limitations

Top

Mathematical Description

The output of the Power Function block can be expressed as following:

$$y(k) = x(k)^{\exp}$$

where y(k) and x(k) are the output and input signal at the simulation time step k, respectively; and exp is the value of the exponent.

Top

Netlist Syntax

COUPL pow ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (ts := @ts , input := @input , exp := @exp) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:-:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value[Unit]
exp	Exponent	real	1.0
ts	Sample Time	real	

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Value	Output	real
input	Input Signal	Input	real

Top

Example

This example illustrates the output of Power Function Math Block.

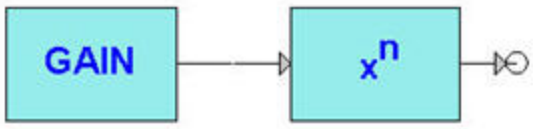


Figure 2. Application example of the VHDL-AMS Power Function Block

Table 3. System Parameters

Component	Parameter	Value [unit]

Power Function Math Block pow1	ts	0 [S]
	input	gain3.val
	exp	2
Gain Block gain3	ts	0
	input	t+0.1
	kp	1

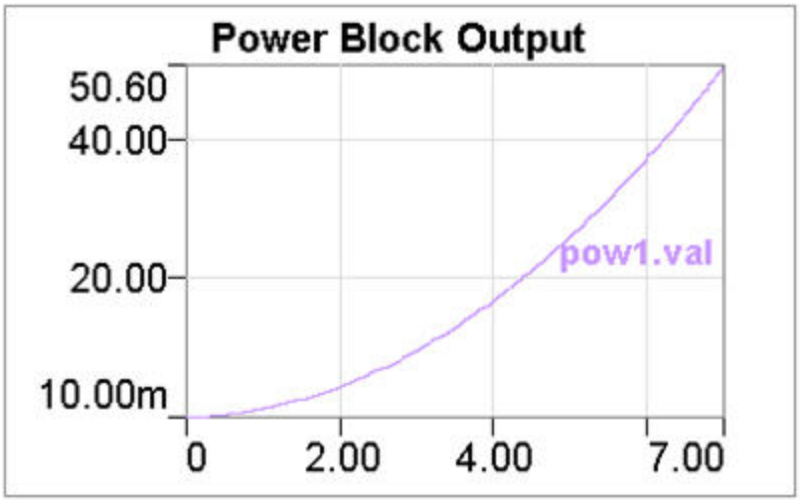


Figure 3. Simulation results-output from Power Function block.

Top

References

Root

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

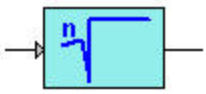


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Parameters

• Input/Output Quantities

• Example

• References

Description

The block computes the n-th radical of the block input signal and provides the result at the block output. The radicand (input) must be a positive number.

To define the radical exponent, enter a numerical value, a variable, or expression in the parameter list.

The sample time is a static value. Enter a numerical value in the parameter list. If '0' is specified, the simulator-defined variable sample time is used. The sample time must be greater than HMIN.

Top

Assumptions and Limitations

Top

Mathematical Description

The output of the Root Function block can be expressed as following:

$$y(k) = \sqrt[root]{x(k)}$$

where y(k) and x(k) are the output and input signal at the simulation time step k, respectively; and root is the value of the radicand.

Top

Netlist Syntax

COUPL rootn ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (ts := @ts , input := @input , root := @root) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:-:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value[Unit]
root	Radical Exponent	real	1.0
ts	Sample Time	real	0.0

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Value	Output	real
input	Input Signal	Input	real

Top

Example

This example illustrates the output of Root Function Block.

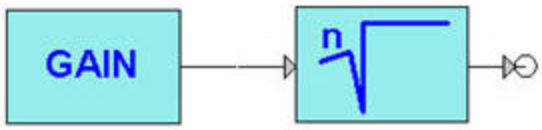


Figure 2. Application example of the VHDL-AMS Root Function Block

Table 3. System Parameters

Component	Parameter	Value [unit]

Root Function Math Block rootn1	ts	0 [S]
	input	gain3.val
	root	2
Gain Block gain3	ts	0
	input	t+0.1
	kp	1

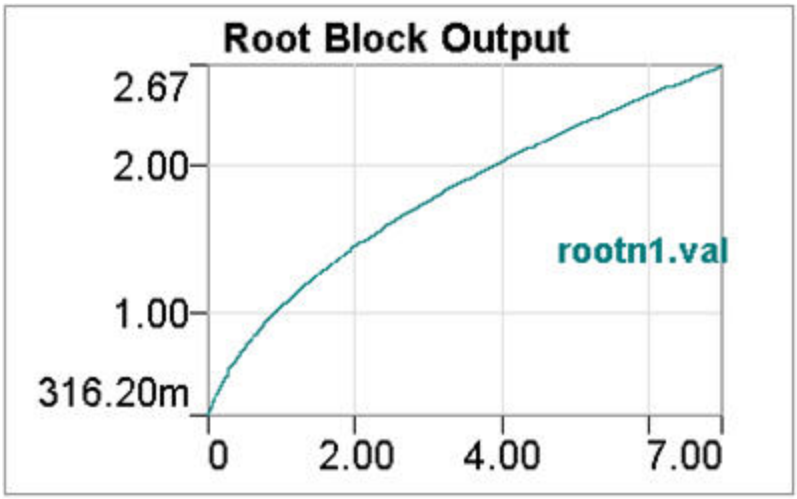


Figure 3. Simulation results-output from Root Function block.

Top

References

Sign

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

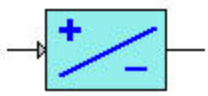


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Parameters

• Input/Output Quantities

• Example

• References

Description

The block provides the discretized signal at the block output, depending on the sign of the block input signal and the adaption factor.

To define the amplitude, enter a numerical value, a variable, or expression in the parameter list.

The sample time is a static value. Enter a numerical value in the parameter list. If '0' is specified, the simulator-defined variable sample time is used. The sample time must be greater than HMIN.

Top

Assumptions and Limitations

Top

Mathematical Description

The output of the Sign Function block can be expressed as following:

$$y(k) = \begin{cases} ampl & if\ x(k) \geq 0 \\ -ampl & if\ x(k) < 0 \end{cases}$$

where y(k) and x(k) are the output and input signal at the simulation time step k, respectively. AMPL is the amplitude of the output.

Top

Netlist Syntax

COUPL sign ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (ts := @ts , input := @input , ampl := @ampl) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:-:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value [Unit]
ampl	Absolute Value of Output Amplitude	real	1.0
ts	Sample Time	real	0.0

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Value	Output	real
input	Input Signal	Input	real

Top

Example

This example illustrates the output of Sign Function Math Block.

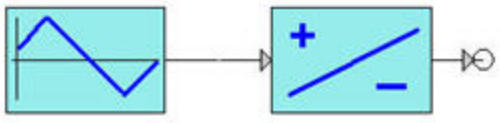


Figure 2. Application example of the VHDL-AMS Sign Function Block

Table 3. System Parameters

Component	Parameter	Value [unit]
Sign Function Math Block sign1	ts	0 [S]
	input	triang1.val
	ampl	2
Triangular Wave Function triang1	freq	1 [Hz]
	ampl	1
	tdelay	0 [S]

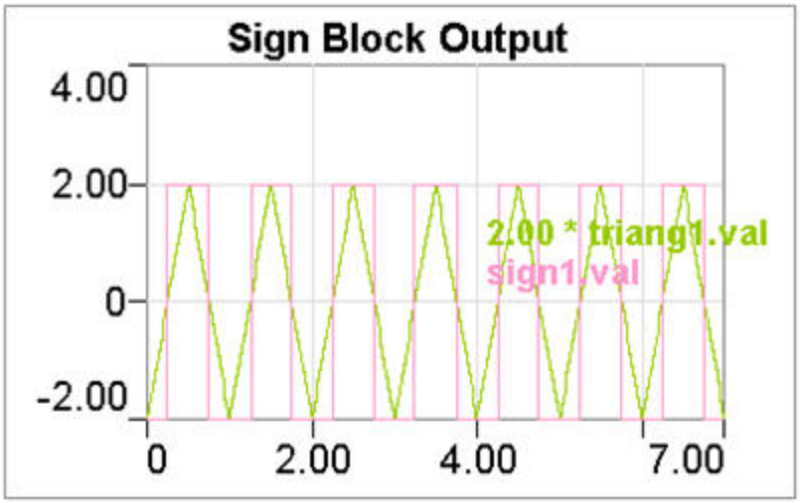


Figure 3. Simulation results-output from the Sign Function block.

Top

References

Signal Processing Blocks

- Comparator Block in VHDL-AMS (comp)
- Limiter Block in VHDL-AMS (lmt)
- Maximum Input Block in VHDL-AMS (max)
- Minimum Input Block in VHDL-AMS (min)
- Multiplier Block in VHDL-AMS (mul)
- Negator Block in VHDL-AMS (neg)
- Summation Block in VHDL-AMS (sum)
- Two-Point Element with Hysteresis in VHDL-AMS (tph)

Comparator

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

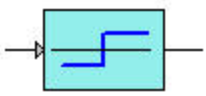


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Parameters

• Input/Output Quantities

• Example

• References

Description

The block represents a two-point element without hysteresis. To define value 1 and 2, and the threshold value, enter a numerical value, a variable, or expression in the parameter list.

The sample time is a static value. Enter a numerical value in the parameter list. If '0' is specified, the simulator-defined variable sample time is used. The sample time must be greater than HMIN.

Top

Assumptions and Limitations

Top

Mathematical Description

The output of the Multiplier block can be expressed as following:

$$y(k) = \begin{cases} val1 & if\ x < thres \\ val2 & if\ x \geq thres \end{cases}$$

where *y(k)* and *x* are the output and input signal at the simulation time step *k*, respectively; *thres* is the threshold.

Top

Netlist Syntax

COUPL comp ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (ts := @ts , input := @input , val1 := @val1 , val2 := @val2 , thres := @thres) DST: SIM(Type:=SimVHDL) SRC: DB (File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value[Unit]
thres	Threshold T	real	0.0
val1	Value 1	real	1.0
val2	Value 2	real	-1.0
ts	Sample Time	real	0.0

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Value	Output	real
input	Input Signal	Input	real

Top

Example

This example illustrates the output of Comparator Signal Processing Block.

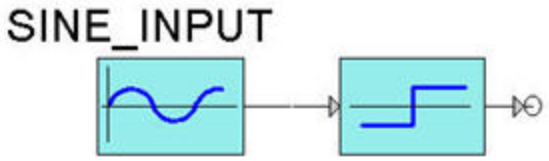


Figure 2. Application example of the VHDL-AMS Comparator Block

Table 3. System Parameters

Component	Parameter	Value [unit]
Comparator Signal Processing Block comp1	ts	0 [S]
	input0	SINE_ INPUT.val
	val1	1
	val2	-1
	thres	3
Sine Block SINE_INPUT	ampl	5
	freq	50 [Hz]
	phase	0 [Deg]

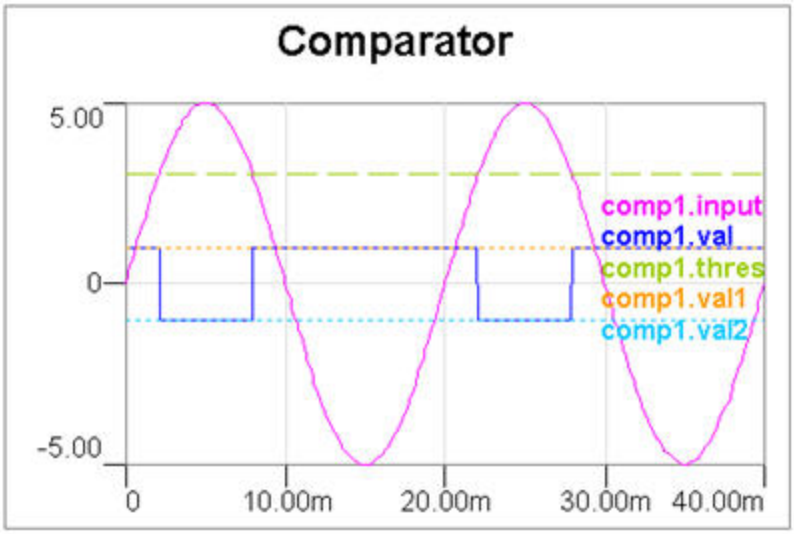


Figure 3. Simulation results-input signal and output from source block.

Top

References

Limiter

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Parameters

• Input/Output Quantities

• Example

• References

Description

The block represents a limitation of the input signal within defined limits. To define the upper and lower limit value, enter a numerical value, a variable, or expression in the parameter list.

The sample time is a static value. Enter a numerical value in the parameter list. If '0' is specified, the simulator-defined variable sample time is used. The sample time must be greater than HMIN.

Top

Assumptions and Limitations

Top

Mathematical Description

The output of the Limiter block can be expressed as following:

$$y(k) = \begin{cases} ul & if\ x(k) \geq ul \\ ll & if\ x(k) \leq ll \\ x(k) & else \end{cases}$$

where y(k) and x(k) are the output and input signal at the simulation time step k, respectively; ul and ll are the upper and lower limit, respectively.

Top

Netlist Syntax

COUPL lmt ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (ts := @ts , input := @input , ul := @ul , ll := @ll) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:-:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value [Unit]
ul	Upper Limit of Output Signal	real	0.0
ll	Lower Limit of Output Signal	real	0.0
ts	Sample Time	real	0.0

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Value	Output	real
input	Input Signal	Input	real

Top

Example

This example illustrates the output of Limiter Signal Processing Block.

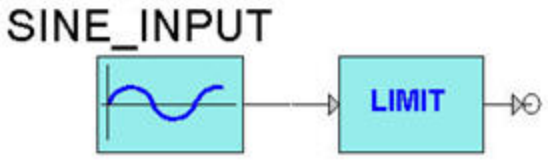


Figure 2. Application example of the VHDL-AMS Limiter Block

Table 3. System Parameters

Component	Parameter	Value [unit]
Limiter Signal Processing Block lmt1	ts	0 [S]
	input0	SINE_ INPUT.val
	ul	1
	ll	-1
Sine Block SINE_INPUT	ampl	5
	freq	50 [Hz]
	phase	0 [Deg]

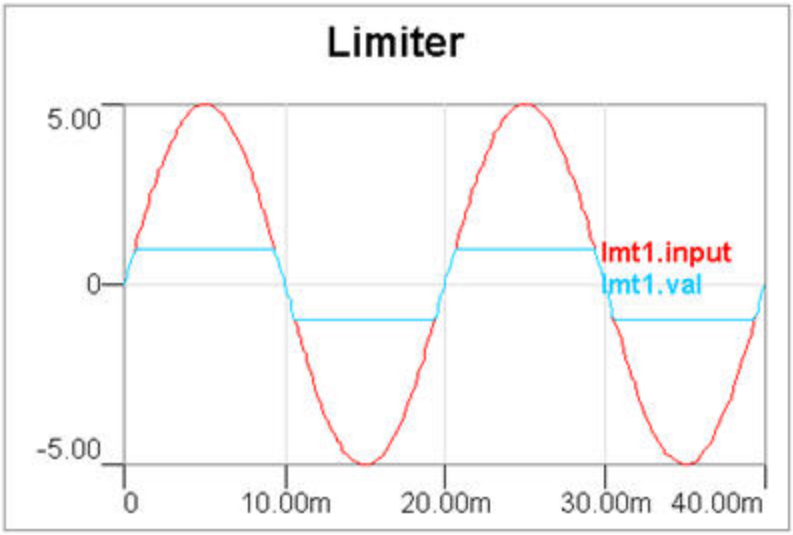


Figure 3. Simulation results-input signal and output of the Limiter block.

Top

References

Maximum of Input Signals

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

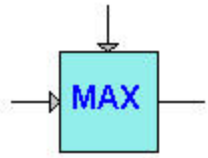


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Parameters

• Input/Output Quantities

• Example

• References

Description

The block provides the maximum value from two input signals at each time step at the block output. The sample time is a static value. Enter a numerical value in the parameter list. If '0' is specified, the simulator-defined variable sample time is used. The sample time must be greater than HMIN.

Top

Assumptions and Limitations

Top

Mathematical Description

The output of the Maximum of the Input Signal function can be expressed as following:

$$y(k) = Max(x_1(k), x_2(k), \cdots, x_n(k))$$

where y(k) is the output signal at the simulation time step k, and xi(k) is the ith input signal at the simulation time step k.

Top

Netlist Syntax

COUPL max ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (ts := @ts , input0 := @input0 , input1 := @input1) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value[Unit]
ts	Sample Time	real	0.0

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Value	Output	real
input0/input1	Input Signal 0/Input Signal 1	Input	real

Top

Example

This example illustrates the output of Maximim Signal Processing Block.

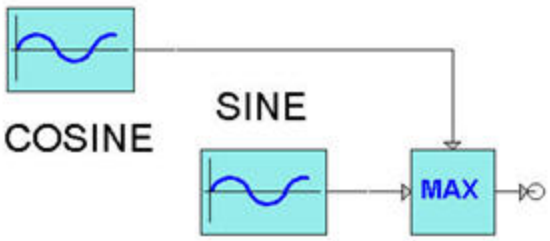


Figure 2. Application example of the VHDL-AMS Maximum Block

Table 3. System Parameters

Component	Parameter	Value [unit]
Maximum Signal Processing Block max1	ts	0 [S]
	input0	SINE.val
	input1	COSINE.val
Sine Block SINE	ampl	5
	freq	50 [Hz]
	phase	0 [Deg]
Cosine Block COSINE	ampl	5
	freq	50 [Hz]
	phase	90 [Deg]

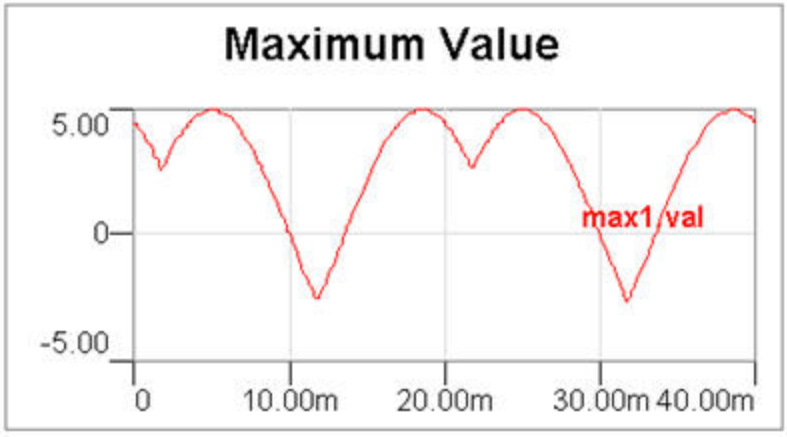


Figure 3. Simulation results-output from the Maximum of Input Signals block.

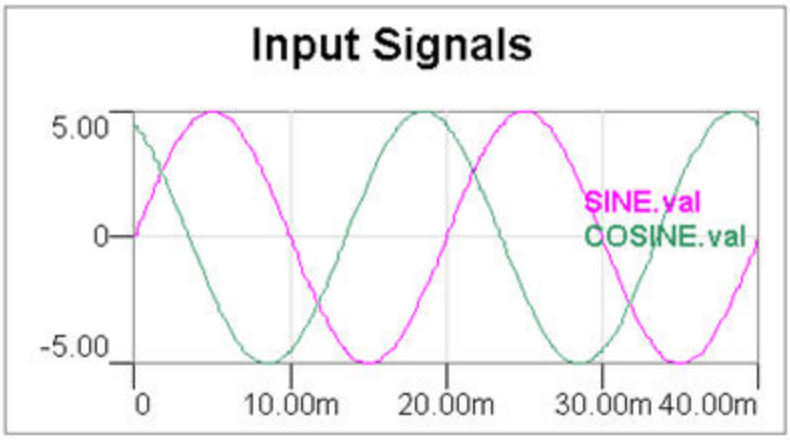


Figure 3. Simulation results-inputs to the Maximum of Input Signals block.

Top

References

Minimum of Input Signals

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

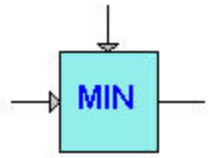


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Parameters

• Input/Output Quantities

• Example

• References

Description

The block provides the minimum value from two input signals at each time step at the block output.

The sample time is a static value. Enter a numerical value in the parameter list. If '0' is specified, the simulator-defined variable sample time is used. The sample time must be greater than HMIN.

Top

Assumptions and Limitations

Top

Mathematical Description

The output of the Minimum of the Input Signal function can be expressed as following:

$$y(k) = Min(x_1(k), x_2(k), \cdots, x_n(k))$$

where y(k) is the output signal at the simulation time step k, and xi(k) is the ith input signal at the simulation time step k.

Top

Netlist Syntax

COUPL min ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (ts := @ts , input0 := @input0 , input1 := @input1) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value[Unit]
ts	Sample Time	real	0.0

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Value	Output	real
input0/input1	Input Signal 0/Input Signal 1	Input	real

Top

Example

This example illustrates the output of Minimum Signal Processing Block.

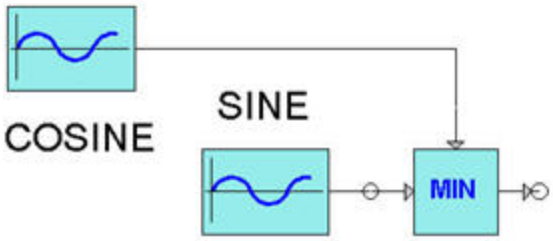


Figure 2. Application example of the VHDL-AMS Minimum Block

Table 3. System Parameters

Component	Parameter	Value [unit]
Minimum Signal Processing Block min1	ts	0 [S]
	input0	SINE.val
	input1	COSINE.val
Sine Block SINE	ampl	5
	freq	50 [Hz]
	phase	0 [Deg]
Cosine Block COSINE	ampl	5
	freq	50 [Hz]
	phase	90 [Deg]

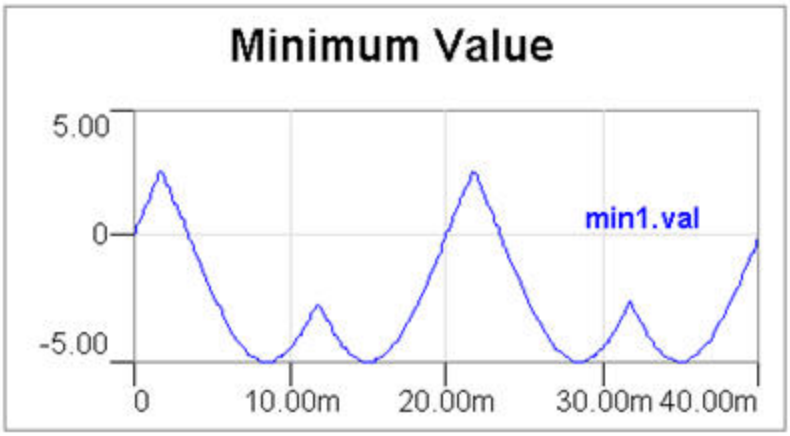


Figure 3. Simulation results-output from the Minimum of Input Signals block.

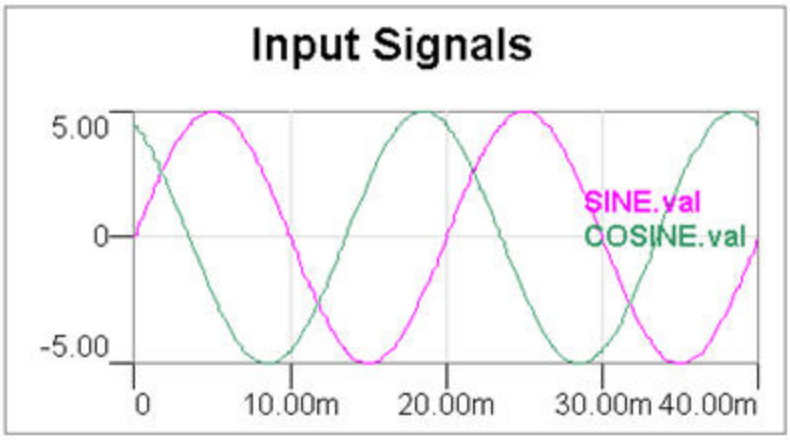


Figure 4. Simulation results-inputs to the Minimum of Input Signals block.

Top

References

Multiplier

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

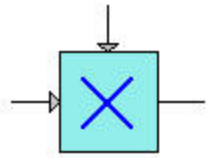


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Parameters

• Input/Output Quantities

• Example

• References

Description

The block multiplies two input signals at each time step and provides the result at the block output.

The sample time is a static value. Enter a numerical value in the parameter list. If '0' is specified, the simulator-defined variable sample time is used. The sample time must be greater than HMIN.

Top

Assumptions and Limitations

Top

Mathematical Description

The output of the Multiplier block can be expressed as following:

$$y(k) = x_1(k) \cdot x_2(k) \cdot \ldots \cdot x_n(k))$$

where y(k) is the output signal at the simulation time step k, and xi(k) is the ith input signal at the simulation time step k.

Top

Netlist Syntax

COUPL mul ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (ts := @ts , input0 := @input0 , input1 := @input1) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value[Unit]
ts	Sample Time	real	0.0

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Value	Output	real
input0/input1	Input Signal/Input Signal 1	Input	real

Top

Example

This example illustrates the output of Multiply Signal Processing Block.

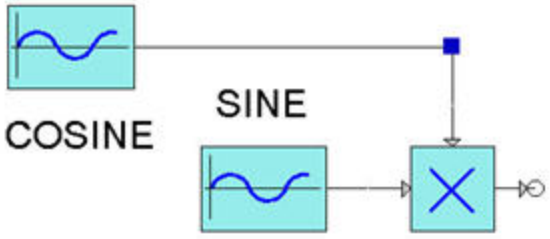


Figure 2. Application example of the VHDL-AMS Multiply Block

Table 3. System Parameters

Component	Parameter	Value [unit]
Multiply Signal Processing Block mul1	ts	0 [S]
	input0	SINE.val
	input1	COSINE.val
Sine Block SINE	ampl	5
	freq	50 [Hz]
	phase	0 [Deg]
Cosine Block COSINE	ampl	5
	freq	50 [Hz]
	phase	90 [Deg]

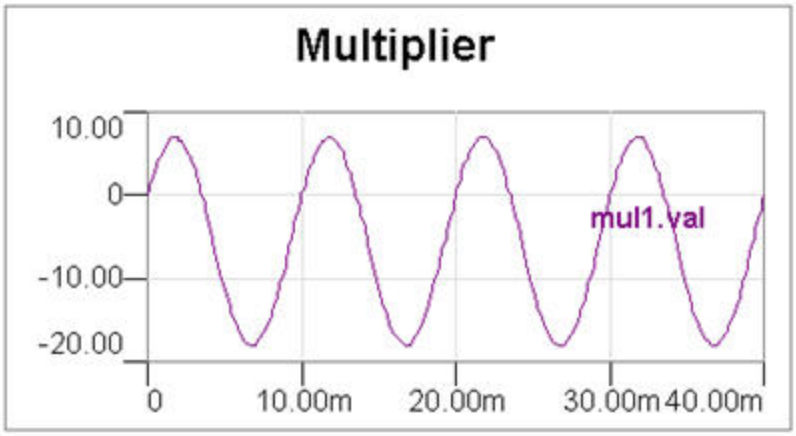


Figure 3. Simulation results-output from the Multiplier block.

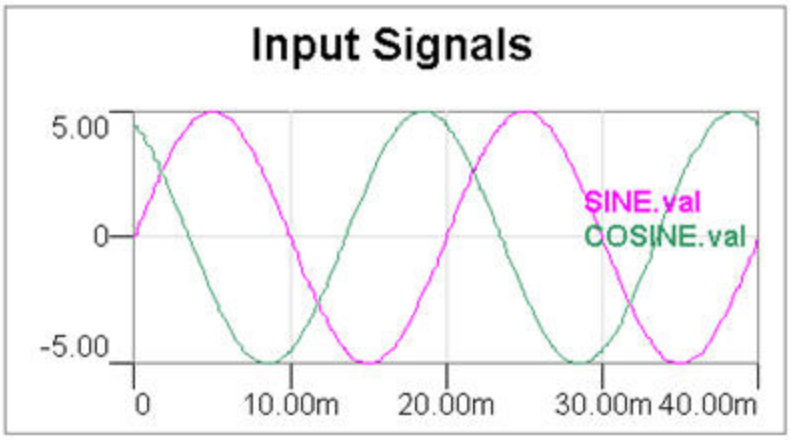


Figure 4. Simulation results-inputs to the Multiplier block.

Top

References

Negator

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Parameters

• Input/Output Quantities

• Example

• References

Description

The block multiplies the input signal at each time step with -1 and provides the result at the block output.

The sample time is a static value. Enter a numerical value in the parameter list. If '0' is specified, the simulator-defined variable sample time is used. The sample time must be greater than HMIN.

Top

Assumptions and Limitations

Top

Mathematical Description

The output of the Negator block can be expressed as following:

$$y(k) = -x(k)$$

where y(k) and x(k) are the output and input signal at the simulation time step k, respectively.

Top

Netlist Syntax

COUPL neg ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (ts := @ts , input := @input) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:-:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value[Unit]
ts	Sample Time	real	0.0

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Value	Output	real
input	Input Signal	Input	real

Top

Example

This example illustrates the output of Negator Signal Processing Block.

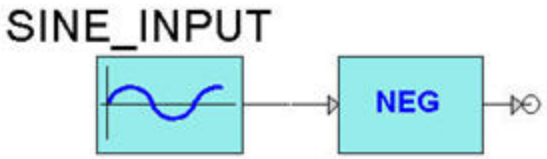


Figure 2. Application example of the VHDL-AMS Negator Block

Table 3. System Parameters

Component	Parameter	Value [unit]
Negator Signal Processing Block neg1	ts	0 [S]
	input0	SINE_ INPUT.val

Sine Block SINE_INPUT	ampl	5
	freq	50 [Hz]
	phase	0 [Deg]

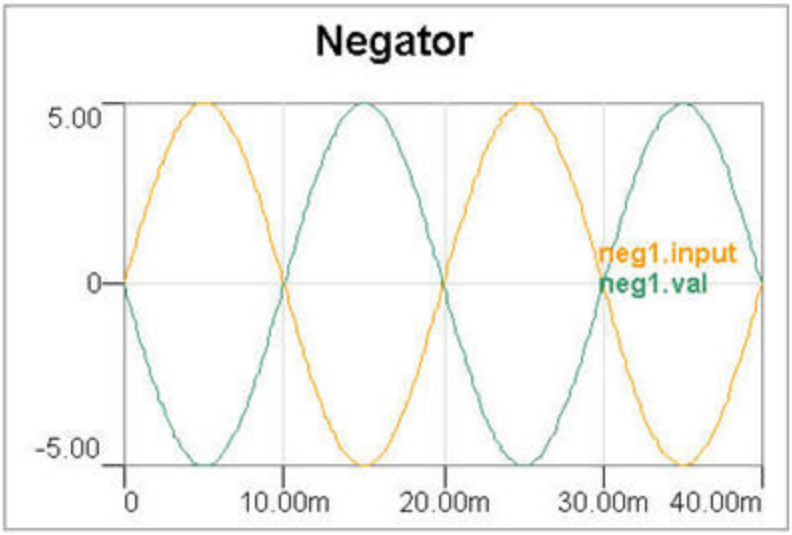


Figure 3. Simulation results-input signal and output of the Negator block.

Top

References

Summation

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

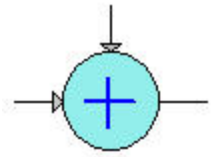


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Parameters

• Input/Output Quantities

• Example

• References

Description

The block adds up two input signals at each time step and provides the result at the block output.

The sample time is a static value. Enter a numerical value in the parameter list. If '0' is specified, the simulator-defined variable sample time is used. The sample time must be greater than HMIN.

Top

Assumptions and Limitations

Top

Mathematical Description

The output of the Summation block can be expressed as following:

$$y(k) = x_1(k) + x_2(k) + \cdots + x_n(k))$$

where y(k) is the output signal at the simulation time step k, and xi(k) is the ith input signal at the simulation time step k.

Top

Netlist Syntax

COUPL sum ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (ts := @ts , input0 := @input0 , input1 := @input1) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value[Unit]
ts	Sample Time	real	0.0

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Value	Output	real
input0/input1	Input Signal 0/Input Signal 1	Input	real

Top

Example

This example illustrates the output of Summation Signal Processing Block.

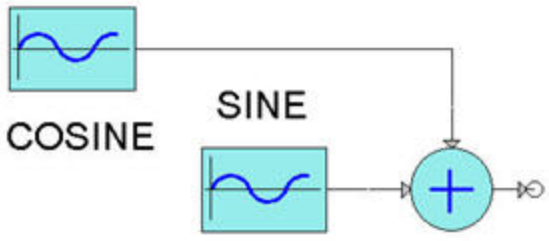


Figure 2. Application example of the VHDL-AMS Summation Block

Table 3. System Parameters

Component	Parameter	Value [unit]

Summation Signal Processing Block sum1	ts	0 [S]
	input0	SINE.val
	input1	COSINE.val
Sine Block SINE	ampl	5
	freq	50 [Hz]
	phase	0 [Deg]
Cosine Block COSINE	ampl	5
	freq	50 [Hz]
	phase	90 [Deg]

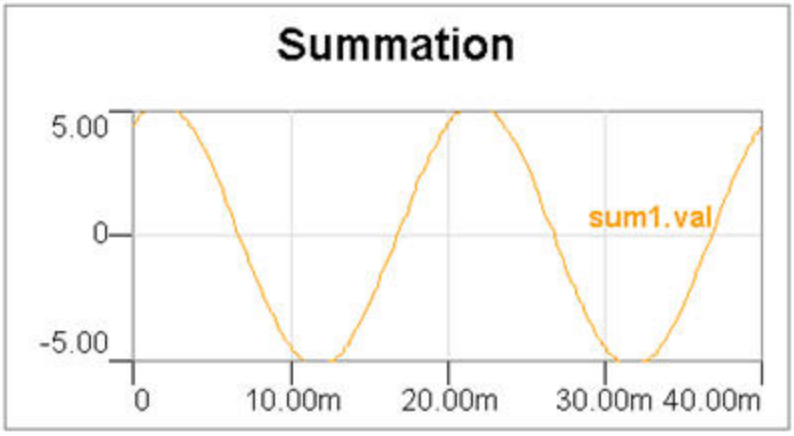


Figure 3. Simulation results-output from the Summation block.

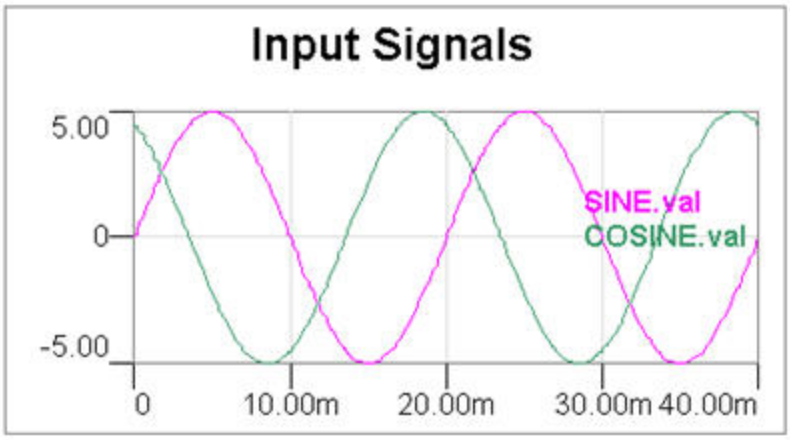


Figure 3. Simulation results-inputs to the Summation block.

Top

References

Two-point Element with Hysteresis

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

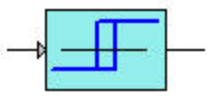


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Parameters

• Input/Output Quantities

• Example

• References

Description

The block represents a two-point element with hysteresis.

To define value 1 and 2, threshold 1 and 2, and the initial value for output, enter a numerical value, a variable, or expression in the parameter list.

The sample time is a static value. Enter a numerical value in the parameter list. If '0' is specified, the simulator-defined variable sample time is used. The sample time must be greater than HMIN.

Top

Assumptions and Limitations

Top

Mathematical Description

The output of the Two-Point Element with Hysteresis block can be expressed as following:

$$y(k) = \begin{cases} val1 & if\ x(k) < thres1 \\ val1 & if\ x(k) < thres2\ and\ x(k-1) = thres1 \\ val2 & if\ x(k) > thres1\ and\ x(k-1) = thres2 \\ val2 & if\ x(k) > thres2 \end{cases}$$

where y(k) and x(k) are the output and input signal at the simulation time step k, respectively; thres1 and thres2 are the thresholds.

Top

Netlist Syntax

COUPL tph ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (y0 := @y0 , ts := @ts , input := @input , val1 := @val1 , val2 := @val2 , thres1 := @thres1 , thres2 := @thres2) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:-:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value[Unit]
thres1	Threshold T1	real	0.0
thres2	Threshold T2	real	0.0
val1	Value 1	real	1.0
val2	Value 2	real	-1.0
y0	Initial Value	real	0.0
ts	Sample Time	real	0.0

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Value	Output	real
input	Input Signal	Input	real

Top

Example

This example illustrates the output of Two-Point with Hysteresis Signal Processing Block.

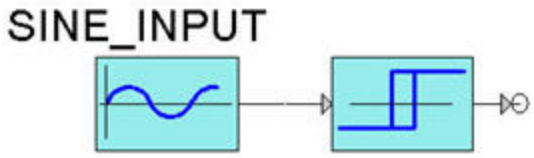


Figure 2. Application example of the VHDL-AMS Two-Point with Hysteresis Block

Table 3. System Parameters

Component	Parameter	Value [unit]
Two-Point with Hysteresis Signal Processing Block tph1	ts	0 [S]
	input0	SINE_ INPUT.val
	val1	1
	val2	-1
	thres1	-2
	thres2	2
Sine Block SINE_INPUT	ampl	5
	freq	50 [Hz]
	phase	0 [Deg]

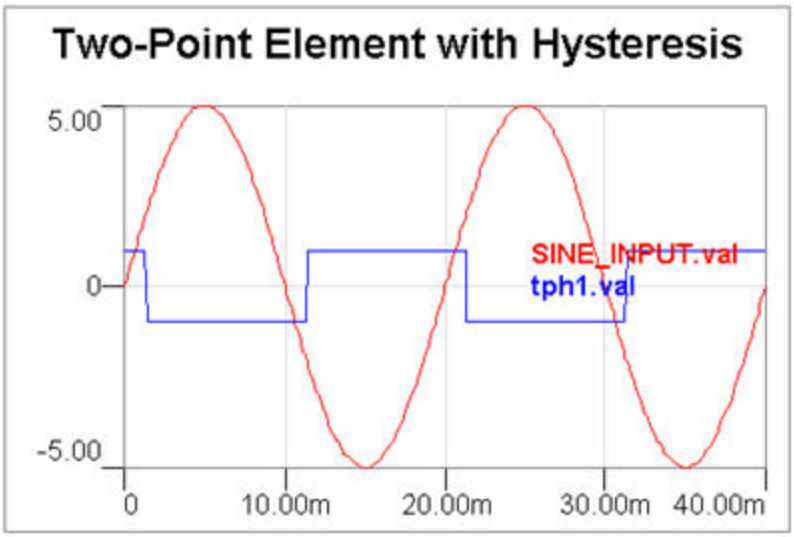


Figure 3. Simulation results-input signal and output from the Two-point Element with Hysteresis block.

Top

References

Source Blocks

- Constant Value Block in VHDL-AMS (const)
- Random Block in VHDL-AMS (random)
- Step Function Block in VHDL-AMS (step)

Constant Value

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Parameters

• Input/Output Quantities

• Example

• References

Description

The block provides a value calculated at each simulation step. To define the constant value, enter a numerical value, a variable, or expression in the parameter list. The sample time is a static value. Enter a numerical value in the parameter list. If '0' is specified, the simulator-defined variable sample time is used. The sample time must be greater than HMIN.

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

COUPL const ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (ts := @ts , const := @const , ac_mag := @ac_mag , ac_phase := @ac_phase) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value [Unit]
const	Constant Value	real	1.0
ts	Sample Time	real	0.0 [s]
ac_mag	Magnitude of Input (AC)	real	1.0e-3
ac_ phase	Phase Shift of Input (AC)	real	0.0

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Value	Output	real

Top

Example

This example illustrates the output of Constant Source Block.

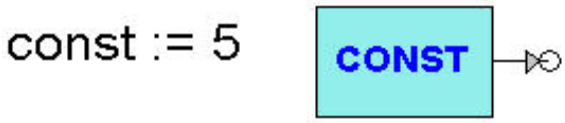


Figure 2. Application example of the VHDL-AMS Constant Source Block

Table 3. System Parameters

Component	Parameter	Value [unit]
Constant Source Block const1	ts	0 [S]
	const	5

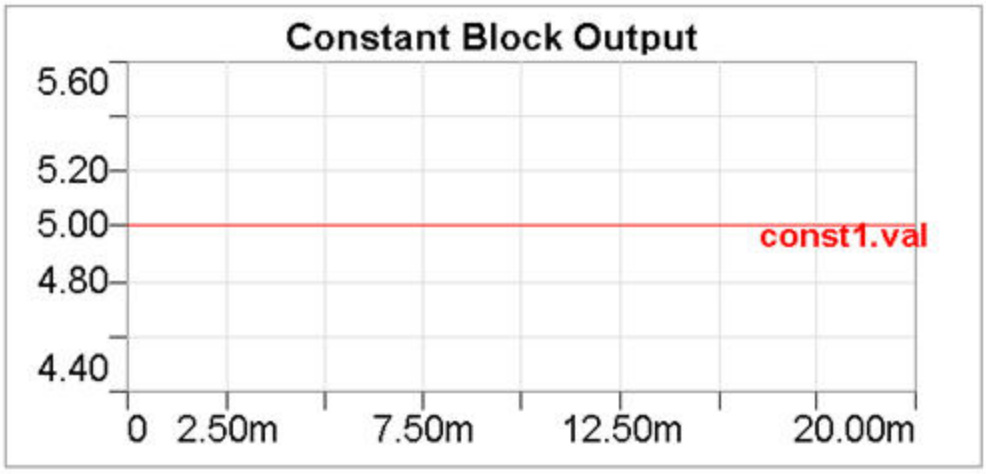


Figure 3. Simulation results-output from Constant source block.

Top

References

Random

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Parameters

• Input/Output Quantities

• Example

• References

Description

The block provides to each simulation step a random number within the defined range of values at the block output. To define the maximum and minimum value, enter a numerical value, a variable, or expression in the parameter list.

The sample time is a static value. Enter a numerical value in the parameter list. If '0' is specified, the simulator-defined variable sample time is used. The sample time must be greater than HMIN.

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

COUPL random ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (ts := @ts , max := @max , min := @min , ac_mag := @ac_mag , ac_phase := @ac_phase) DST: SIM(Type:-:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value [Unit]
max	Maximum Random Value	real	1.0
min	Minimum Random Value	real	-1.0
ts	Sample Time	real	0.0
ac_mag	Magnitude of Input (AC)	real	1.0e-3
ac_ phase	Phase Shift of Input (AC)	real	0.0

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Value	Output	real

Top

Example

This example illustrates the output of Random Source Block.

Figure 2. Application example of the VHDL-AMS Random Source Block

Table 3. System Parameters

Component	Parameter	Value [unit]
Random Source Block const1	ts	0 [S]
	max	2
	min	-2

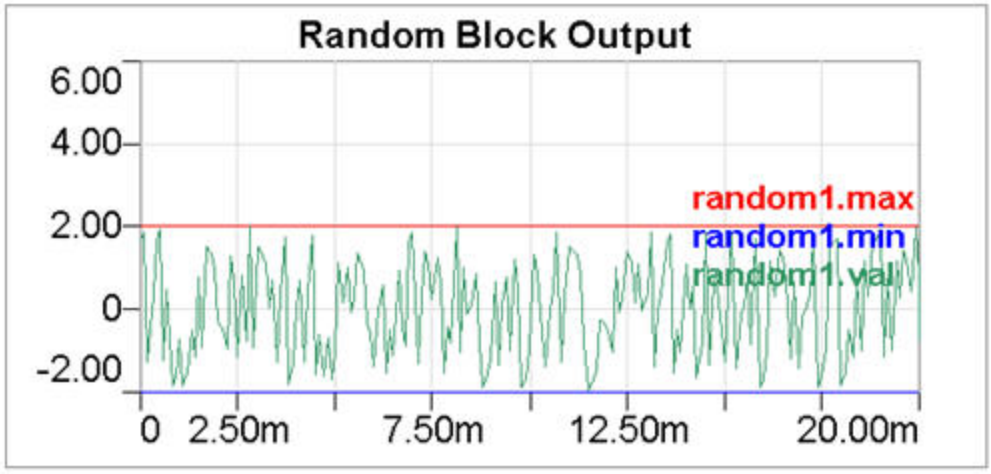


Figure 3. Simulation results-input and output of the Random source block.

Top

References

Step Function

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

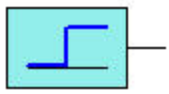


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Parameters

• Input/Output Quantities

• Example

• References

Description

The block provides a signal that changes at time t=t0 instantly from an initial value y0 to the specified amplitude value k. To define the amplitude value and initial delay, enter a numerical value, a variable, or expression in the parameter list. The initial delay is set only once at the start of the simulation.

The sample time is a static value. Enter a numerical value in the parameter list. If '0' is specified, the simulator-defined variable sample time is used. The sample time must be greater than HMIN.

Top

Assumptions and Limitations

Top

Mathematical Description

The output of the Step Function can be expressed as following:

$$y(k) = y0 \qquad for\ t < t0$$
$$y(k) = ampl \quad for\ t \geq t0$$

where y(k) is the output signal at simulation time step k.

Top

Netlist Syntax

COUPL step ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (t0 := @t0 , ts := @ts , y0 := @y0 , ampl := @ampl , ac_mag := @ac_mag , ac_phase := @ac_phase) DST: SIM(Type:-:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value [Unit]
t0	Step Time	real	0.0
ampl	Amplitude	real	1.0
ts	Sample Time	real	0.0
ac_mag	Magnitude of Input (AC)	real	1.0e-3
ac_ phase	Phase Shift of Input (AC)	real	0.0
y0	Initial Value	real	0.0

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Value	Output	real

Top

Example

This example illustrates the output of Step Function Source Block.

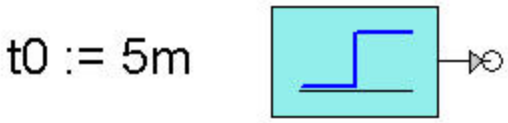


Figure 2. Application example of the VHDL-AMS Step Function Source Block

Table 3. System Parameters

Component	Parameter	Value [unit]
Step Function Source Block step1	t0	5m [S]
	ampl	5

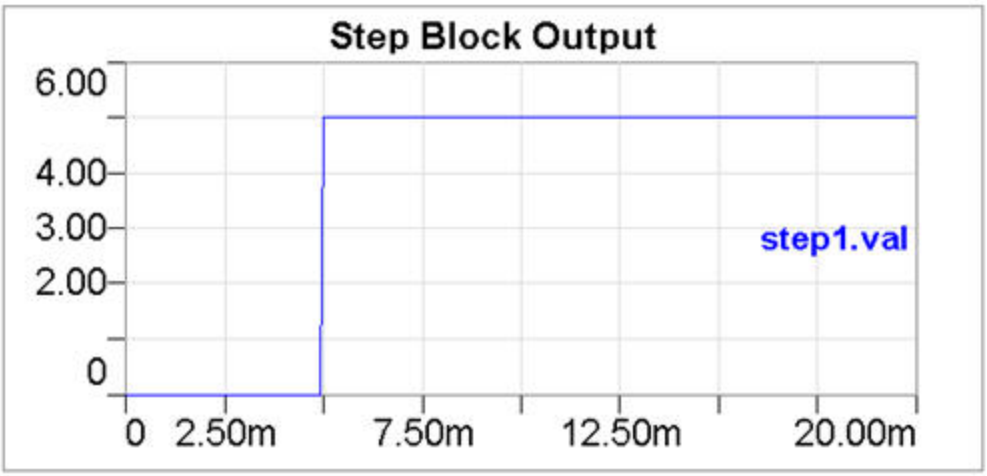


Figure 3. Simulation results-output from the Step Function source block.

Top

References

Modeling with Circuit Components

Electrical domain components are defined by an internal identifier, the name, their conservative nodes and a set of parameters. Terminals of conservative nodes cannot be deleted on the sheet. Electrical domain components appear in the *Circuit* folder of the **Basic Elements VHDLAMS** library. Select the folder *Circuit*, choose one of the groups, click a name, drag the component onto the sheet and release the mouse button.

All forms of linear and nonlinear components can be used to model electrical circuits. However, only electrically reasonable circuits lead to correct simulation results. Voltage sources, current sources and switches are ideal components. Each independent circuit must be connected to a ground node at least once.

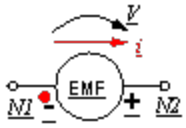
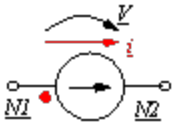
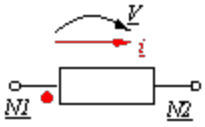
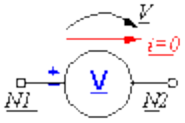
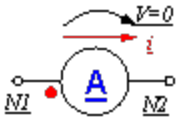
The *Circuit* folder provides components in the following categories:

- Electrical Machines
- Passive Elements
- Semiconductor System Level
- Sources
- Switches
- Transformers

See also Network Configurations

Load Reference Arrow System of Circuit Components

The counting direction of current and voltage is marked by the red point or the plus sign at the symbol of electrical components.

	Voltage Sources	Current Sources	Passive Components
Electrical			
	Voltmeter	Ammeter	Wattmeter
Meters			

Network Configurations

In Twin Builder only ammeters are allowed as controlling components for current controlled elements. These must be inserted properly in the controlling branch. If sources are part of mutual controlling sources in the circuit, stability problems may occur if the total gain of the loop is greater than or equal to one.

The following types of network configurations are invalid:

• Series connection of ideal current sources

• Series connection of inductors and ideal current sources

• Series connection of inductors with different initial values of current, I01¹I02

• Series connection of an inductor with an initial current value and an opened ideal switch or non-conducting system level semiconductor

• Parallel connection of ideal voltage sources

• Parallel connection of capacitors with ideal voltage sources

• Parallel connection of capacitors with different initial values of voltage, V01¹V02

• Meshes which consist only of ideal sources (short-circuit)

• Open-ended branches

Electrical Machines

- DC Machine Electrical Excitation (dcme)
- DC Machine Permanent Magnet Excitation (dcmp)
- Induction Machine (im)
- Synchronous Machine Electrical Excitation without Damper (syme)
- Synchronous Machine Electrical Excitation with Damper (symed)
- Synchronous Machine Permanent Magnet Excitation without Damper (symp)
- Synchronous Machine Permanent Magnet Excitation with Damper (sympd)

DC Machine Linear Electrical Excitation

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

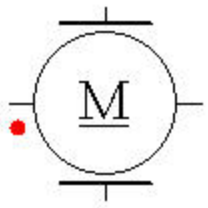


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Input/Output Quantities

• Example

• References

Description

The model represents a DC machine as a lumped circuit component. By proper connection of armature and excitation circuit, separately excited, series and shunt machines can be modeled. To define the parameter values, enter a numerical value, a variable, or expression in the corresponding field of parameter list.

The equation system is implemented on condition of a linear magnetic circuit. Index ***a*** represents the armature circuit quantities, the index ***e*** the quantities of excitation circuit.

Top

Assumptions and Limitations

Model Limits of DC Machine Models

• The nonlinear magnetic circuit (DC machine with Nonlinear electrical excitation) is able to consider the dependence on excitation flux and inductance caused by the excitation current.

• Armature and exciter circuit of the DC machine model are considered to be completely decoupled.

• No consideration of saturation effects in the armature q-axis caused by the armature current.

• No consideration of armature reaction on exciting field

• No consideration of eddy-current and hysteresis loss caused by armature rotation and pulsating-current supply system

• Friction losses (parasitic torques) are not considered in the model; they can be added with the load torque parameter externally

Top

Mathematical Description

Top

Netlist Syntax

COUPL dcme ?InstanceName(@InstanceName):(@(Refbase)@(ID)) n1 := %0 , n2 := %1 , e1 := %2 , e2 := %3 (la := @la , le := @le , j := @j , iepsi := @iepsi , ia0 := @ia0 , ie0 := @ie0 , n0 := @n0 , phi0 := @phi0 , load := @load , ra := @ra , re := @re) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
n1	Node N1	electrical
n2	Node N2	electrical
e1	Node Excitation E1	electrical
e2	Node Excitation E2	electrical

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
load	Load Torque	real	0.0 [Nm]
ra	Armature Resistance Sum of resistances of all windings of armarture circuit	real	1.2 [Ohm]
la	Armature Inductance Sum of inductivities of all windings of armarture circuit	real	0.0095 [H]
re	Excitation Resistance Sum of resistances of all windings of excitation circuit	real	1.2 [Ohm]
le	Excitation Inductance Sum of inductivities of all windings of excitation circuit	real	0.0095 [H]
j	Moment of Inertia	real	0.004 [kg*m^2]
iepsi	Excitation Current-Flux Rate Nominal excitation flux-linkage normalized to nominal excitation current	real	1.0 [Vs/A]
ia0	Initial Armature Current	real	0.0 [A]
ie0	Initial Excitation Current	real	0.0 [A]
n0	Initial Rotor Speed	real	0.0 [rpm]
phi0	Initial Rotor Position	real	0.0 [rad]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
n	Rotor Speed [rpm]	Output	real
phi	Rotor Angle [rad]	Output	real

Top

Example

This example demonstrates the setup of a basic DC Electrical Excitation motor with linear excitation and mechanical load.

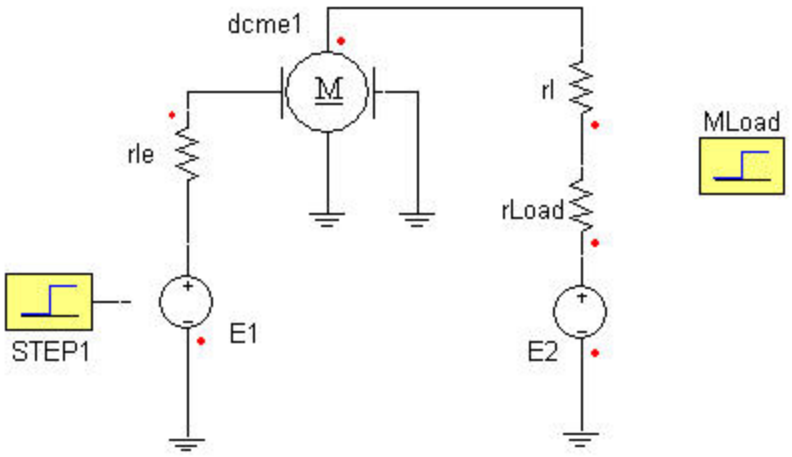


Figure 2. Application examples of the DC Machine Linear Electrical Excitation model.

Table 4. System Parameters

Component	Parameter	Value [unit]
DC Electrical Excitation dcme1	la	4.2m [H]
	j	0.35
	load	MLoad.VAL
	re	244.4 [Ohm]
	ra	0.34 [Ohm]
	Voltage Source E1	cmf T- E- P- 1- - V-

		A- L
Voltage Source E2	emf	90
Step Function STEP1	T0	0 [S]
	ampl	440
Step Function MLoad	T0	0.5 [S]
	ampl	50
Resistor rle/rl	r	1m [Ohm]
Resistor rLoad	r	3m [Ohm]

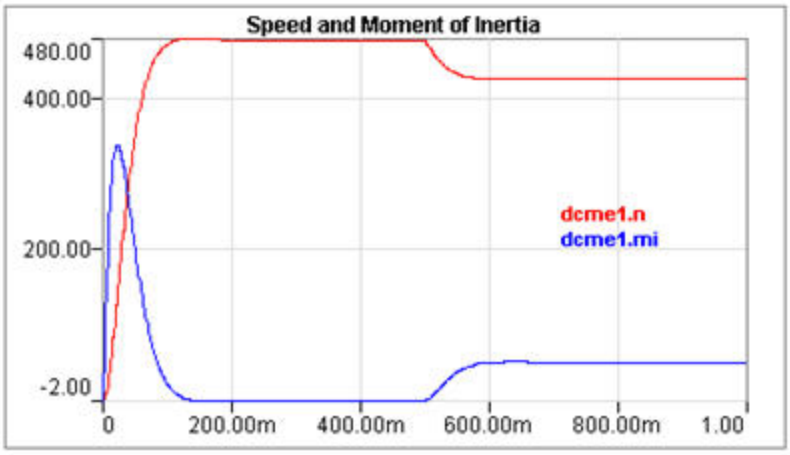


Figure 3. Simulation results-speed and electromagnetic torque of dcme1.

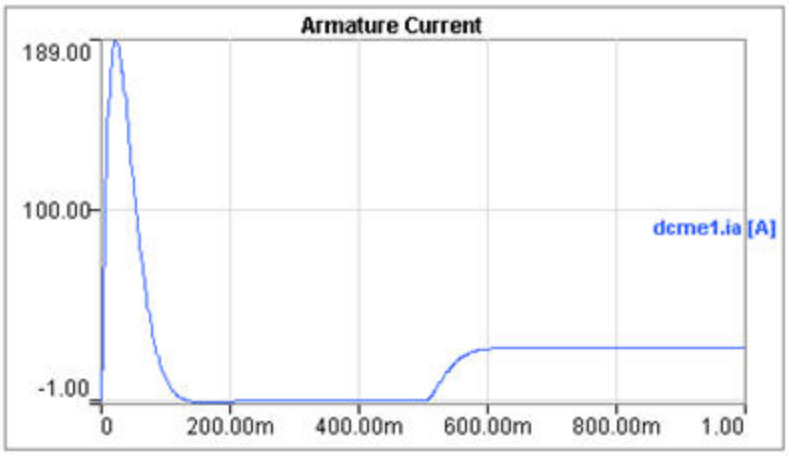


Figure 4. Simulation results-armature current of dcme1.

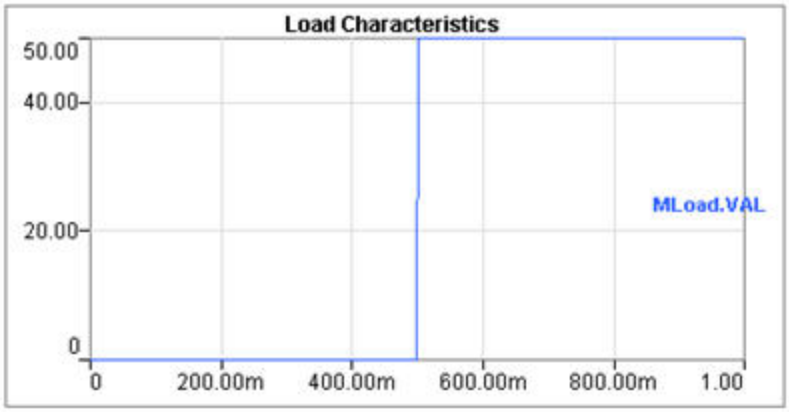


Figure 5. Simulation results-load of dcme1.

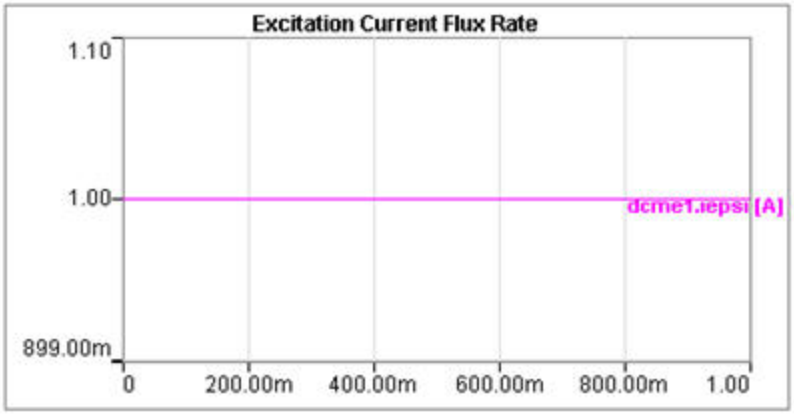


Figure 6. Simulation results-excitation current flux of dcme1.

Top

References

DC Machine Permanent Excitation

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

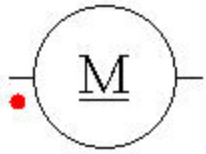


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Input/Output Quantities

• Example

• References

Description

The model represents a DC machine with permanent excitation as a lumped circuit component. To define the parameter values, enter a numerical value, a variable, or expression in the corresponding field of parameter list.

The equation system is implemented on condition of a linear magnetic circuit. Index a represents the armature circuit quantities.

Top

Assumptions and Limitations

Model Limits of DC Machine Models

• The nonlinear magnetic circuit (DC machine with Nonlinear electrical excitation) is able to consider the dependence on excitation flux and inductance caused by the excitation current.

• Armature and exciter circuit of the DC machine model are considered to be completely decoupled.

• No consideration of saturation effects in the armature q-axis caused by the armature current.

• No consideration of armature reaction on exciting field

• No consideration of eddy-current and hysteresis loss caused by armature rotation and pulsating-current supply system

• Friction losses (parasitic torques) are not considered in the model; they can be added with the load torque parameter externally

Top

Mathematical Description

Top

Netlist Syntax

COUPL dcmp ?InstanceName(@InstanceName):(@(Refbase)@(ID)) n1 := %0 , n2 := %1 (la := @la , ke := @ke , j := @j , ia0 := @ia0 , n0 := @n0 , phi0 := @phi0 , load := @load , ra := @ra) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:-:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
n1	Node N1	electrical
n2	Node N2	electrical

Top

Parameters

Table 2

Name	Description	Data Type	Default Value[Unit]
load	Load Torque	real	0.0 [Nm]
ra	Armature Resistance Sum of resistances of all windings of armarture circuit	real	1.0 [Ohm]
la	Armature Inductance Sum of inductivities of all windings of armarture circuit	real	0.01 [H]
ke	Motor Constant	real	1.0
j	Moment of Inertia	real	0.075 [kg*m^2]
ia0	Initial Current Stator Phase a	real	0.0 [A]
n0	Initial Rotor Speed	real	0.0 [rpm]
phi0	Initial Rotor Position	real	0.0 [rad]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
n	Rotor Speed [rpm]	Output	real
phi	Rotor Angle [rad]	Output	real

Top

Example

This example demonstrates the use of a DC Permanent Magnet motor being driven by a switched DC supply. The circuit in Figure 2 shows the supply, switching transistor, and motor hookup. Figure 3 shows the block diagram of the control circuit to sample the motor speed and create the TR_Control signal used to drive the gate of the transistor to provide the proper average voltage to hold the motor at speed.

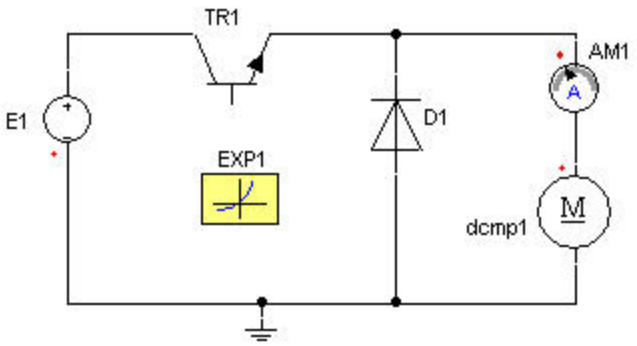


Figure 2. Application examples of the VHDL-AMS DC Permanent Magnet Motor model.

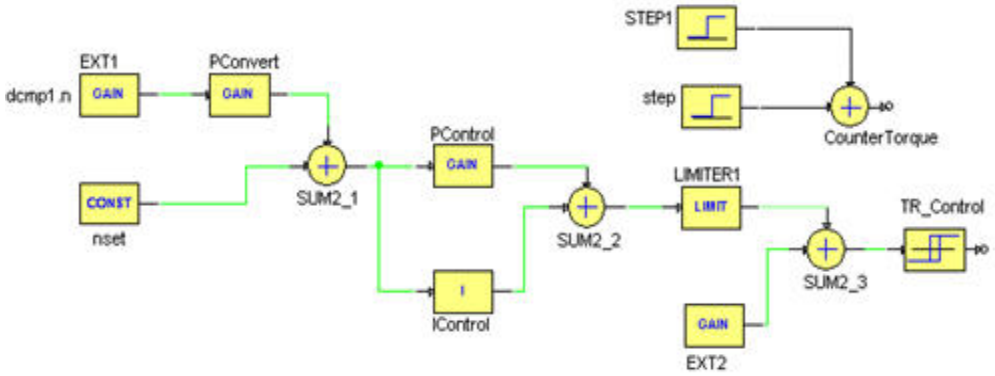


Figure 3. Block Diagram of the control logic.

Table 4. System Parameters

Component	Parameter	Value [unit]
Voltage Source e1	emf	0.9k [V]
BJT Transistor TR1	CH	EXP1
	CTRL	TR_Control.VAL
Diode D1	CH	EXP1
Exponential Function EXP1	VT	35m [V]
	ISAT	1p [A]

	RR	0.1Meg [Ohm]
DC Permanent Magnet Excitation dcmp1	la	9.5m [H]
	j	0.004
	load	CounterTorque.VAL
	ke	0.544
	ra	1.2 [Ohm]
Step Function step	T0	0.05 [S]
	ampl	9
Step Function STEP1	T0	0.0 [S]
	ampl	1
SUM CounterTorque	Input[0]	step.VAL
	Input[1]	STEP1.VAL
GAIN EXT1	INPUT	dcmp1.n
	KP	1
GAIN PConvert	INPUT	EXT1.VAL
	KP	-16.67m
CONST nset	TS	0
	CONST	16.67
SUM SUM2_1	Input[0]	nset.VAL
	Input[1]	PConvert.VAL
GAIN PControl	INPUT	SUM2_1.VAL

	KP	0.15k
INTG IControl	INPUT	SUM2_1.VAL
	KI	20
	Y0	0
	CTRL	0
SUM SUM2_2	Input[0]	IControl.VAL
	Input[1]	PControl.VAL
LIMIT limiter1	Input	SUM2_2.VAL
	UL	20
	LL	0
GAIN EXT2	INPUT	-(AM1.I)
	KP	1
SUM SUM2_3	Input[0]	EXT2.VAL
	Input[1]	LIMITER1.VAL
TPH TR_Control	Input	SUM2_3.VAL
	THRES1	-2.5
	THRES2	2.5
	VAL1	-1
	VAL2	1

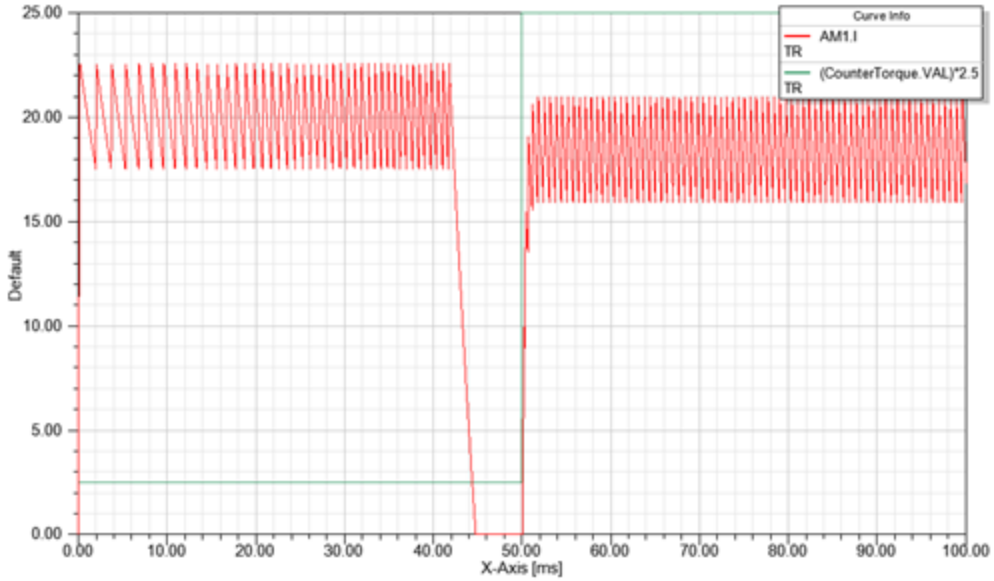


Figure 4. Simulation results-load torque and the current through dcmp1

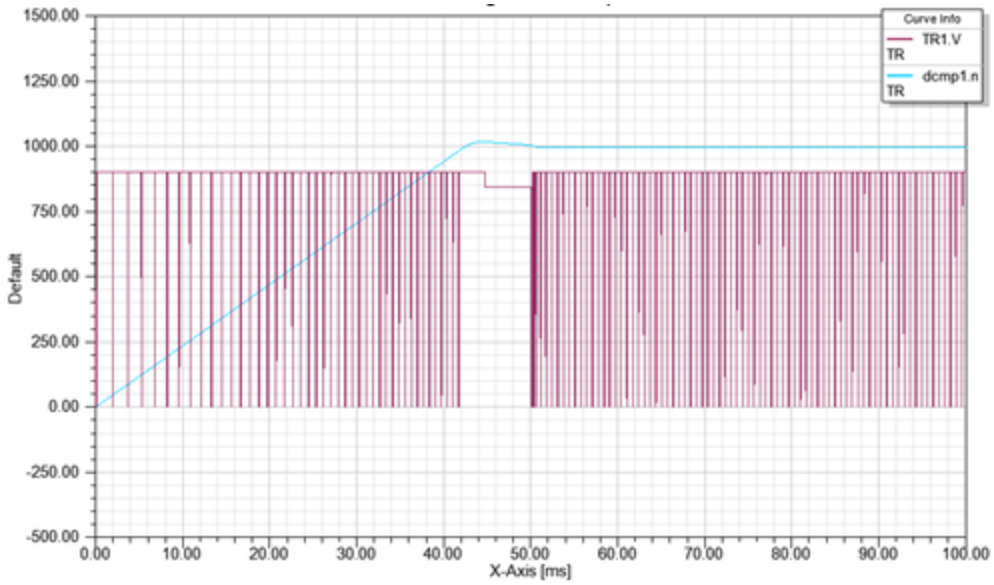


Figure 5. Simulation results-speed and voltage applied to dcmp1

Top

References

Induction Machine with Squirrel Cage Rotor

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

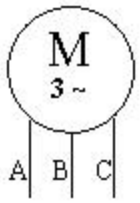


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Input/Output Quantities

• Example

• References

Description

The model represents an induction machine with squirrel cage rotor and star-connected stator wingdings as a lumped circuit component. The circuit nodes A – B – C are the terminals of star-connected stator windings. To define the parameter values, enter a numerical value, a variable, or expression in the corresponding field of parameter list.

The equation system is implemented in a stator-fixed coordinate system (a-b coordinates). Index 1 represents the stator quantities, index 2 the rotor quantities. The phase quantities of the real three-phase induction machine are indicated with a, b, c.

Note: If the line-to-line voltage v_{ab}, v_{bc}, v_{ac} or the line currents i_a, i_b, i_c of the induction machine are of special interest, voltmeters or ammeters can be connected to the induction machine.

Top

Assumptions and Limitations

Model Limits of Induction Machine Model

• 3-phase symmetrical induction machine with squirrel cage rotor and star-connected stator wingdings without neutral node (no zero phase-sequence system).

• Linear and iron-loss free magnetic circuit.

• No consideration of skin effects in the wingdings (restricted simulation accuracy at e.g. start-up processes; typical case: current-displacement motor connected to the mains)

• Exclusive consideration of fundamental flux linkage between stator and rotor wingdings

• Rotor position-independent leakage inductances.

• Friction losses (parasitic torques) are not considered in the model; they can be added with the load torque parameter externally.

Top

Mathematical Description

Top

Netlist Syntax

COUPL im ?InstanceName(@InstanceName):(@(Refbase)@(ID)) a := %0 , b := %1 , c := %2 (ls1 := @ls1 , ls2 := @ls2 , p := @p , j := @j , i1a0 := @i1a0 , i1b0 := @i1b0 , i1c0 := @i1c0 , i2a0 := @i2a0 , i2b0 := @i2b0 , i2c0 := @i2c0 , n0 := @n0 , phim0 := @phim0 , load := @load , r1 := @r1 , r2 := @r2 , lm := @lm) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
a	Node A	electrical
b	Node B	electrical
c	Node C	electrical

Top

Parameters

Table 2

Name	Description	Data Type	Default Value[Unit]

load	Load Torque	real	0.0 [Nm]
r1	Stator Resistance	real	1.0 [Ohm]
r2	Rotor Resistance	real	0.88 [Ohm]
ls1	Stator Leakage Inductance leakage inductance of stator side	real	0.010 [H]
ls2	Rotor Leakage Inductance leakage inductance of rotor side	real	0.013 [H]
lm	Main Inductance coupling inductance between stator and rotor side	real	0.4 [H]
p	Number of Pole Pairs	real	1.0 [/]
j	Moment of Inertia	real	0.075 [kg*m^2]
i1a0	Initial Current Stator Phase a	real	0.0 [A]
i1b0	Initial Current Stator Phase b	real	0.0 [A]
i1c0	Initial Current Stator Phase c	real	0.0 [A]
i2a0	Initial Current Rotor Phase a with ratio of induction machine convert to stator side	real	0.0 [A]
i2b0	Initial Current Rotor Phase b with ratio of induction machine convert to stator side	real	0.0 [A]
i2c0	Initial Current Rotor Phase c with ratio of induction machine convert to stator side	real	0.0 [A]
n0	Initial Rotor Speed	real	0.0 [rpm]
phi0	Initial Rotor Position	real	0.0 [rad]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
n	Rotor Speed [rpm]	Output	real
phi	Rotor Angle [rad]	Output	real

Top

Example

This example demonstrates the use of an Induction Machine Model. The machine is driven by a three phase source comprised if three voltage sources spaced 120 degrees apart in phase. The load presented to the machine is the sum of a load torque and a generator torque as illustrated in the block diagram of Figure 3, allowing the effects of switching the motor load to be demonstrated.

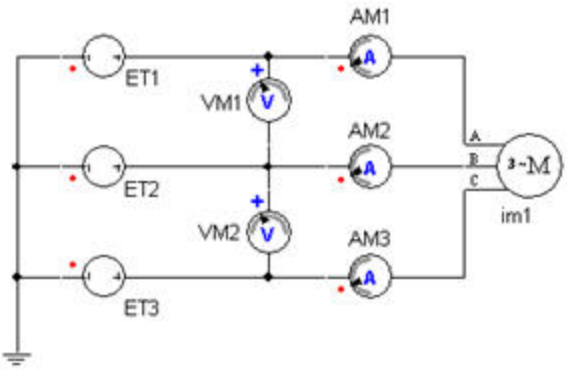


Figure 2. Application examples of the VHDL-AMS Induction Machine model

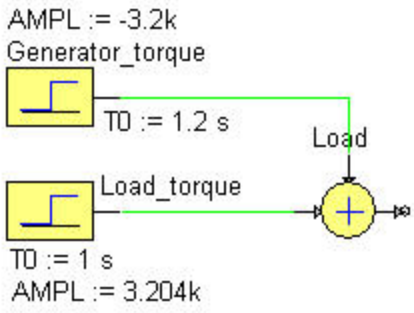


Figure 3. Induction Machine load block diagram

Table 4. System Parameters

Component	Parameter	Value [unit]
Time Controlled Voltage Source (Sinusoidal) ET1/ET2/ET3	freq	50 [Hz]
	tperio	20m [S]
	ampl	0.546k [V]
	phase ET1/ET2/ET3	0/-120/-240 [Deg]
Induction Machine im1	p	2
	j	10.5
	ls1	0.1726m [H]
	ls2	0.20222m [H]
	load	Load.VAL
	r1	4.8m [Ohm]
	r2	13.3m [Ohm]
	lm	9.81m [H]
Step Function (Generator_torque)	ampl	-3.2k
	TO	1.2 [S]
TO	1.0 [S]	
Sum Function (Load)	input[0]	Load_ torque.VAL
	input[1]	Generator_ torque.VAL

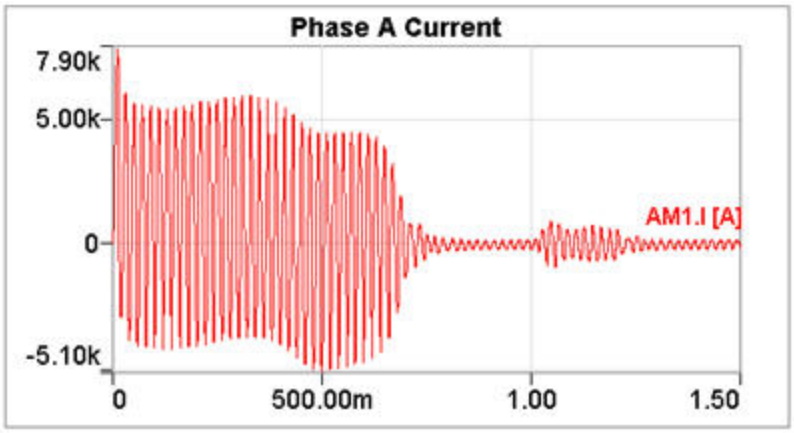


Figure 4. Simulation results-Phase A input current to Induction motor.

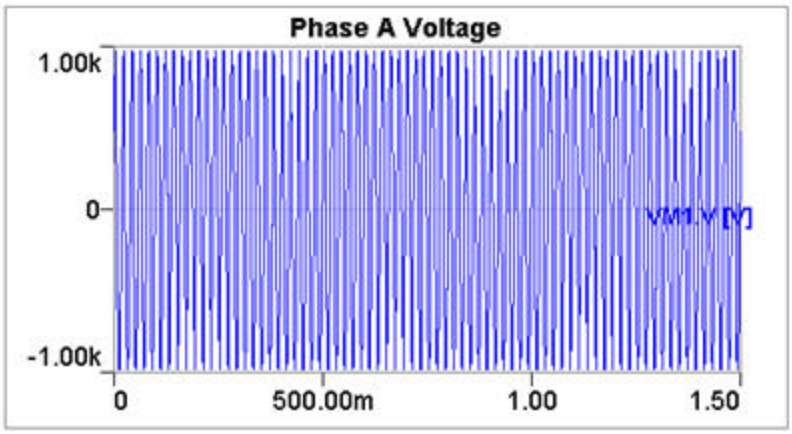


Figure 5. Simulation results-Phase A input voltage to Induction Motor.

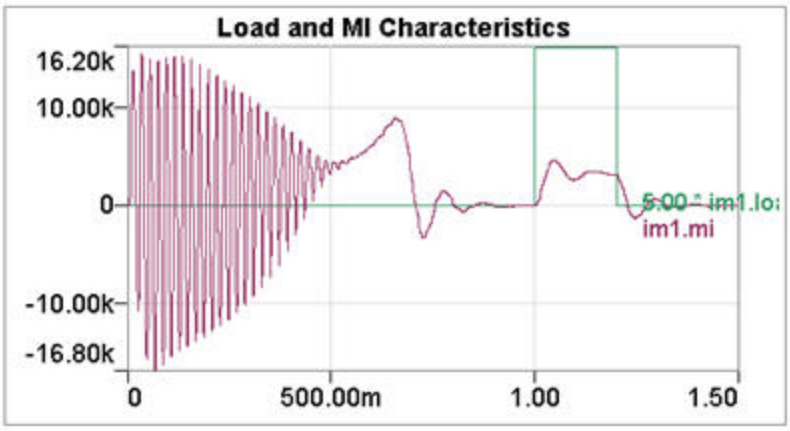


Figure 6. Simulation results-varying load characteristic and electromagnetic torque (im1.mi) developed in the motor.

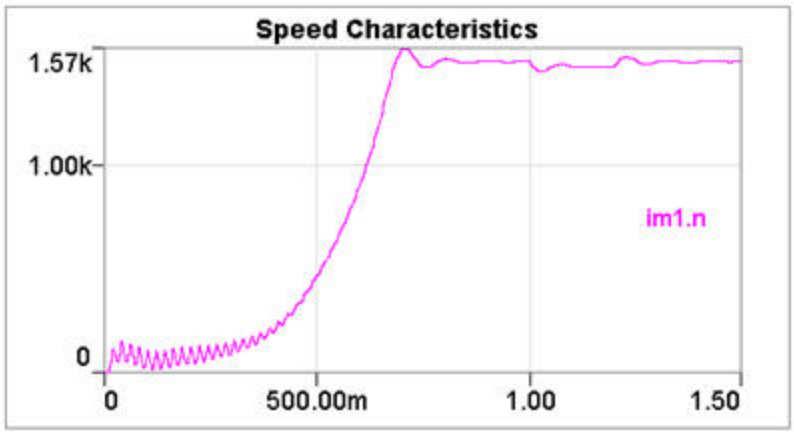


Figure 7. Simulation results-speed characteristic of induction motor (im1) showing run up to steady state and variation due to load change.

Top

References

Synchronous Machine Electrical Excitation without Damper

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

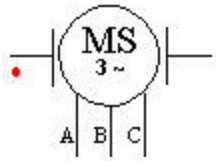


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Input/Output Quantities

• Example

• References

Description

The model represents a linear DC field synchronous machine without damper (internal-field) as a lumped circuit component. Depending on the parameter set, the machine can be operated either as a salient or non-salient pole rotor. The circuit nodes A – B – C are the terminals of star-connected stator winding; the circuit nodes E1 – E2 are the terminals of excitation winding. To define the parameter values, enter a numerical value, a variable, or expression in the corresponding field of parameter list.

The equation system is implemented in a rotor-fixed (rotor-fixed and also rotor flux-fixed) coordinate system (d-q coordinates). Index *1* represents the stator quantities; the index e the quantities of rotor excitation wingdings. The phase quantities of the real three-phase synchronous machine are indicated with a, *b*, *c*.

Note: If the line-to-line voltage v_{ab}, v_{bc}, v_{ac} or the line currents i_a, i_b, i_c of the synchronous machine are of special interest, voltmeters or ammeters can be connected to the synchronous machine.

Top

Assumptions and Limitations

Model Limits of Synchronous Machine Models

• 3-phase symmetrical synchronous machine with internal-field system and star-connected stator winding without neutral node (no zero phase-sequence system).

• Linear and iron-loss free magnetic circuit.

• No consideration of skin effects in the windings.

• Exclusive consideration of fundamental flux linkage between stator, damper and excitation windings.

• Rotor position-independent leakage inductances.

• Friction losses (parasitic torques) are not considered in the model; they can be added with the load torque parameter externally.

Top

Mathematical Description

Top

Netlist Syntax

COUPL syme ?InstanceName(@InstanceName):(@(Refbase)@(ID)) a := %0 , b := %1 , c := %2 , e1 := %3 , e2 := %4 (l1d := @l1d , l1q := @l1q , le := @le , p := @p , j := @j , i1a0 := @i1a0 , i1b0 := @i1b0 , i1c0 := @i1c0 , ie0 := @ie0 , n0 := @n0 , phi0 := @phi0 , load := @load , r1 := @r1 , re := @re , lm1ed := @lm1ed) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
a	Node A	electrical
b	Node B	electrical
c	Node C	electrical
e1	Node Excitation 1	electrical
e2	Node Excitation 2	electrical

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
load	Load Torque	real	0.0 [Nm]
r1	Stator Resistance	real	0.4 [Ohm]
l1d	Stator Inductance d-Axis	real	0.042 [H]
l1q	Stator Inductance q-Axis	real	0.042 [H]
lm1ed	Mutual Inductance stator-exciter d-Axis	real	0.035 [H]
re	Excitation Resistance	real	1.0 [Ohm]
le	Excitation Inductance d-Axis	real	0.050 [H]
p	Number of Pole Pairs	real	2.0 [/]
j	Moment of Inertia	real	0.075 [kg*m^2]
ie0	Initial Excitation Current	real	0.0 [A]
i1a0	Initial Current Stator Phase a	real	0.0 [A]
i1b0	Initial Current Stator Phase b	real	0.0 [A]
i1c0	Initial Current Stator Phase c	real	0.0 [A]
phi0	Initial Rotor Position	real	0.0 [rad]
n0	Initial Rotor Speed	real	0.0 [rpm]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
n	Rotor Speed [rpm]	Output	real
phi	Rotor Angle [rad]	Output	real

Top

Example

This example demonstrates the use of a Synchronous machine configured as an electrical generator.

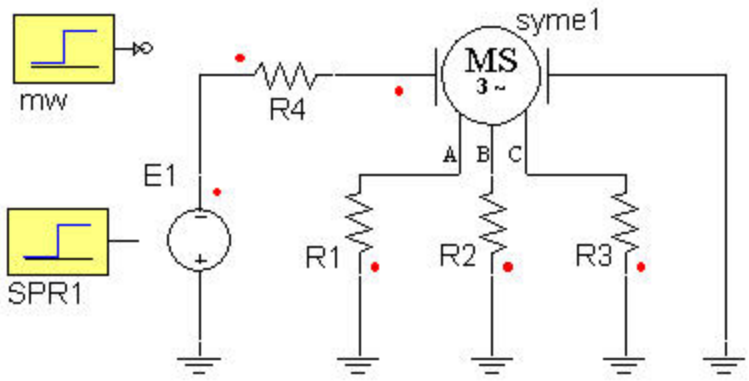


Figure 2. Application example of the VHDL-AMS Synchronous Machine with Electrical Excitation configured as a generator

Table 4. System Parameters

Component	Parameter	Value [unit]
Voltage Source e1	emf	SPR1.VAL
Step Function SPR1	T0	0 [S]
	ampl	500
Step Function mw	T0	0 [S]
	ampl	20
Synchronous Machine (Electrical Excitation without Damping) syme1	l1d	0.04193 [H]
	l1q	0.04193 [H]
	j	0.0129
	load	mw.VAL
	r1	0.410 [Ohm]
Resistor r1/r2/r3	r	15 [Ohm]
Resistor r4	r	20 [Ohm]

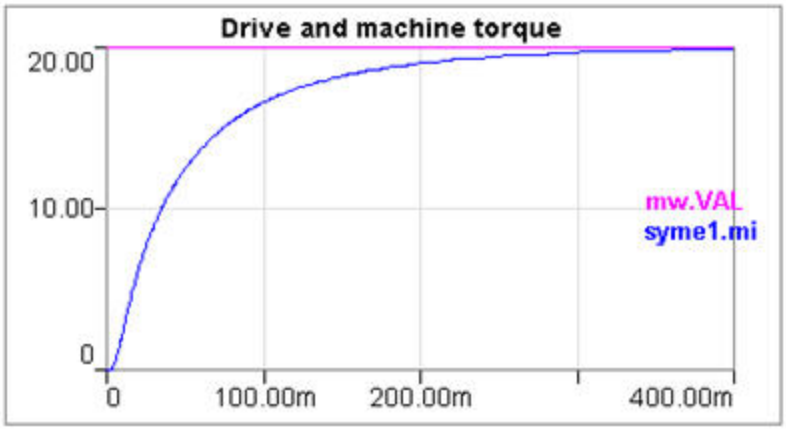


Figure 3. Simulation results-load value (mw.val) and the motor electromagnetic torque (syme1.mi).

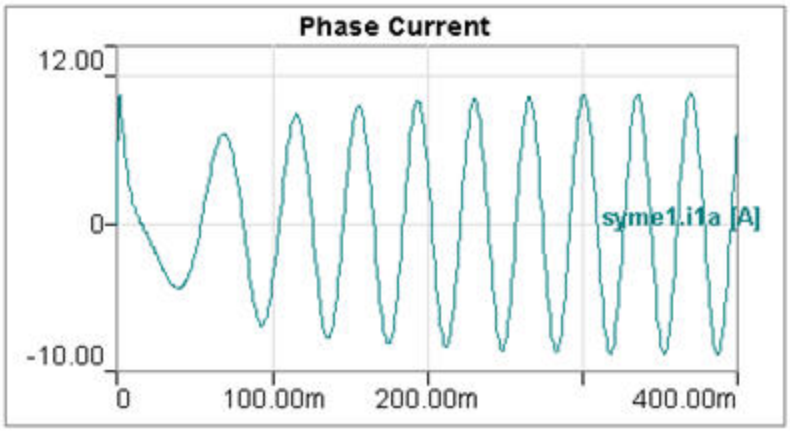


Figure 4. Simulation results-phase A current of syme1.

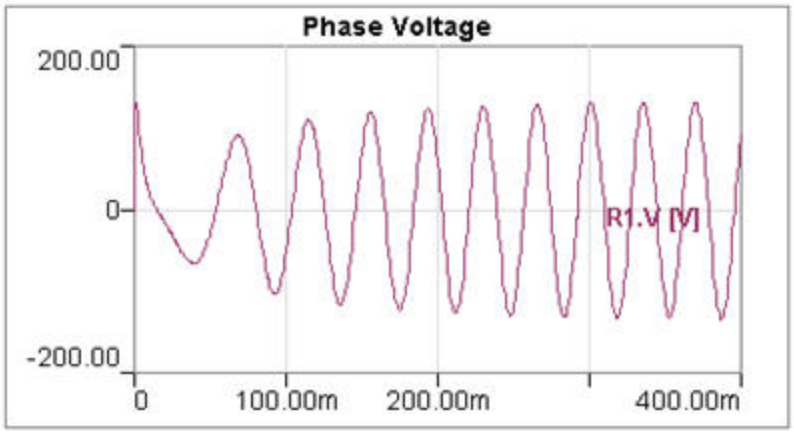


Figure 5. Simulation results-phase A voltage output of syme1.

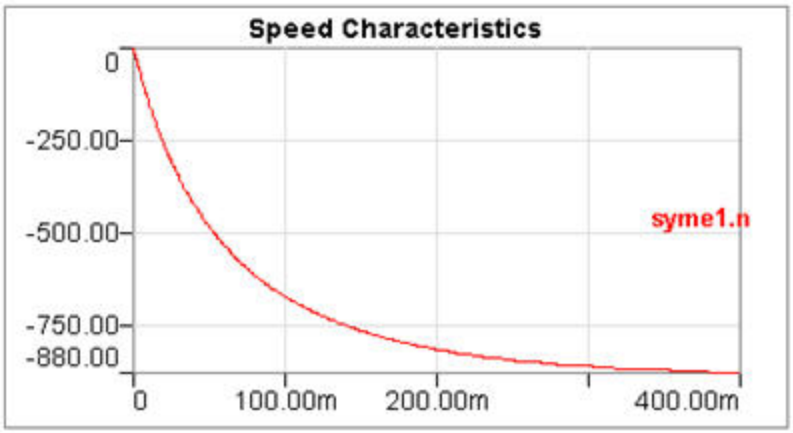


Figure 6. Simulation results-speed of syme1.

Top

References

Synchronous Machine Electrical Excitation with Damper

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

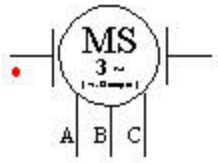


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Input/Output Quantities

• Example

• References

Description

The model represents a linear DC field synchronous machine with damper as a lumped circuit component. The damper circuit is not accessible from outside the machine; the corresponding phase windings are connected at a virtual neutral node. Depending on the parameter set, the machine can be operated either as a salient or non-salient pole rotor. The circuit nodes A – B – C are the terminals of star-connected stator winding; the circuit nodes E1 – E2 are the terminal of excitation winding. To define the parameter values, enter a numerical value, a variable, or expression in the corresponding field of parameter list.

The equation system is implemented in a rotor-fixed (rotor-fixed and also rotor flux-fixed) coordinate system (d-q coordinates). Index 1 represents the stator quantities; the index 2 the equivalent quantities of the damper circuit belonged to the rotor. The phase quantities of the real three-phase synchronous machine are indicated with *a*, *b*, *c*.

In the case of identical parameters for the inductances L1D and L1Q, a synchronous machine with permanent magnet and non-salient pole rotor is modeled. To model a permanent magnet salient-pole machine, the parameters must be different for L1D and L1Q.

Note: If the line-to-line voltage v_{ab}, v_{bc}, v_{ac} or the line currents i_a, i_b, i_c of the synchronous machine are of special interest, voltmeters or ammeters can be connected to the synchronous machine.

Top

Assumptions and Limitations

Model Limits of Synchronous Machine Models

• 3-phase symmetrical synchronous machine with internal-field system and star-connected stator winding without neutral node (no zero phase-sequence system).

• Linear and iron-loss free magnetic circuit.

• No consideration of skin effects in the windings.

• Exclusive consideration of fundamental flux linkage between stator, damper and excitation windings.

• Rotor position-independent leakage inductances.

• Friction losses (parasitic torques) are not considered in the model; they can be added with the load torque parameter externally.

Top

Mathematical Description

Top

Netlist Syntax

COUPL symed ?InstanceName(@InstanceName):(@(Refbase)@(ID)) a := %0 , b := %1 , c := %2 , e1 := %3 , e2 := %4 (l1d := @l1d , l1q := @l1q , l2d := @l2d , l2q := @l2q , lm12d := @lm12d , lm12q := @lm12q , le := @le , lm1ed := @lm1ed , lm2ed := @lm2ed , p := @p , j := @j , i1a0 := @i1a0 , i1b0 := @i1b0 , i1c0 := @i1c0 , i2a0 := @i2a0 , i2b0 := @i2b0 , i2c0 := @i2c0 , ie0 := @ie0 , n0 := @n0 , phi0 := @phi0 , load := @load , r1 := @r1 , r2 := @r2 , re := @re) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:-:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type

a	Node A	electrical
b	Node B	electrical
c	Node C	electrical
e1	Node Excitation E1	electrical
e2	Node Excitation E2	electrical

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
load	Load Torque	real	0.0 [Nm]
r1	Stator Resistance	real	0.4 [Ohm]
re	Excitation Resistance	real	1.0 [Ohm]
r2	Damper Resistance	real	0.24 [Ohm]
l1d	Stator Inductance d-Axis Leakage inductance of stator side	real	0.042 [H]
l1q	Stator Inductance q-Axis Leakage inductance of stator side	real	0.042 [H]
l2d	Damper Inductance d-Axis Leakage inductance of damper circuit side	real	0.042 [H]
l2q	Damper Inductance q-Axis Leakage inductance of damper circuit side	real	0.042 [H]
lm1d	Mutual Inductance stator-damper d-Axis	real	0.041 [H]
lm12q	Mutual Inductance stator-damper q-Axis	real	0.041 [H]
le	Excitation Inductance d-Axis	real	0.050 [H]
lm1ed	Mutual Inductance stator-exciter d-Axis	real	0.035 [H]
lm2ed	Mutual Inductance stator-exciter q-Axis	real	0.030 [H]
p	Number of Pole Pairs	real	2.0 [/]
j	Moment of Inertia	real	0.075 [kg*m^2]

ie0	Initial Excitation Current	real	0.0 [A]
i1a0	Initial Current Stator Phase a	real	0.0 [A]
i1b0	Initial Current Stator Phase b	real	0.0 [A]
i1c0	Initial Current Stator Phase c	real	0.0 [A]
i2a0	Initial Current Damper Phase a	real	0.0 [A]
i2b0	Initial Current Damper Phase b	real	0.0 [A]
i2c0	Initial Current Damper Phase c	real	0.0 [A]
n0	Initial Rotor Speed	real	0.0 [rpm]
phi0	Initial Rotor Position	real	0.0 [rad]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
n	Rotor Speed [rpm]	Output	real
phi	Rotor Angle [rad]	Output	real

Top

Example

This example demonstrates the use of a Synchronous Electrical machine with damper in an basic motor configuration with three phase drive and a switched load.

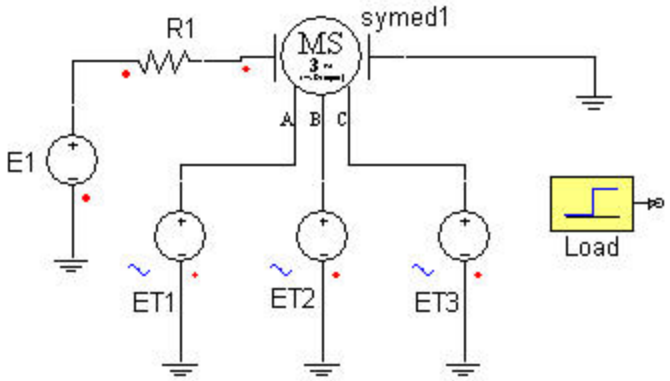


Figure 2. Application examples of the VHDL-AMS Synchronous Electrical Excitation Machine with damping.

Table 4. System Parameters

Component	Parameter	Value [unit]
Time Controlled Voltage Source (Sinusoidal) ET1/ET2/ET3	freq	50 [Hz]
	tperio	20m [S]
	ampl	0..326k [V]
	phase ET1/ET2/ET3	0/-120/-240 [Deg]
Synchronous Electrical Excitation Machine with Damping symed1	p	2
	j	0.075
	load	Load.VAL
	r1	0.4 [Ohm]
	r2	0.24 [Ohm]
	re	1.0 [Ohm]
Step Function Load	T0	0.2 [S]
	ampl	100
Voltage Source E1	emf	40 [V]
Resistor r1	r	1 [Ohm]

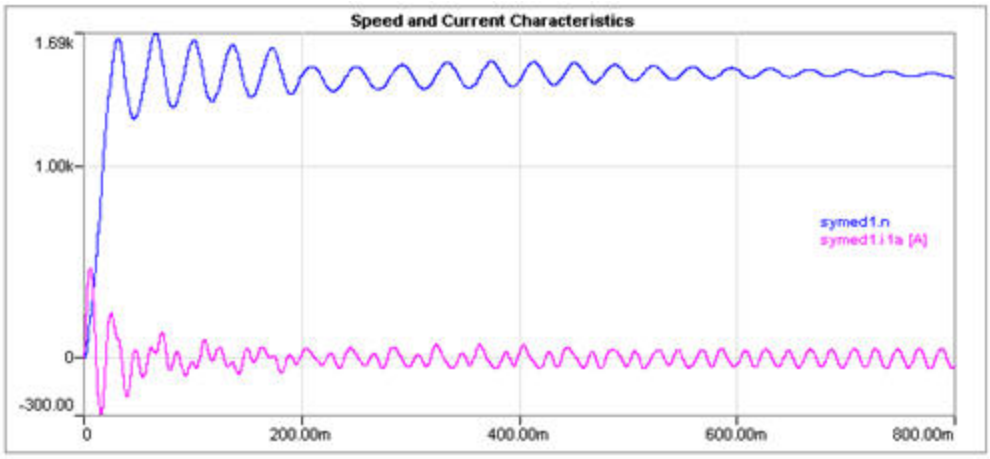


Figure 3. Simulation results-speed and phase A current of symed1.

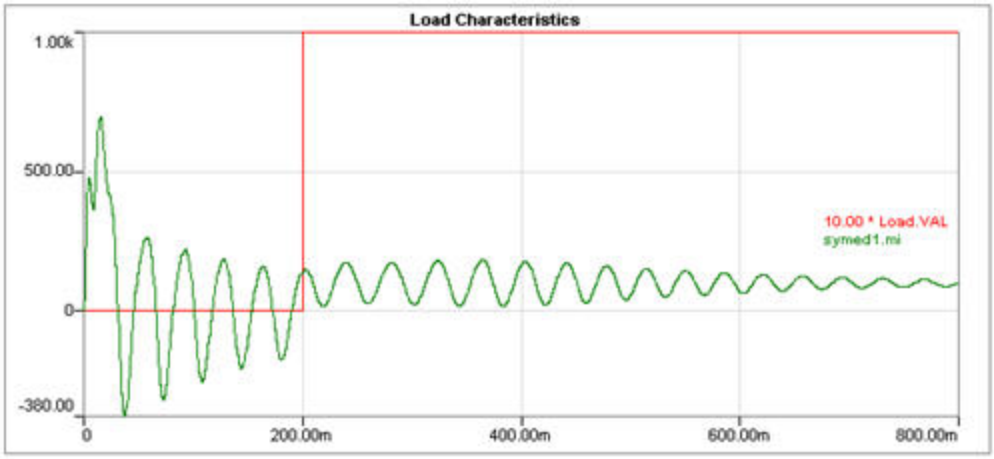


Figure 4. Simulation results-load and electromagnetic torque of symed1.

Top

References

Synchronous Machine Permanent Excitation without Damper

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

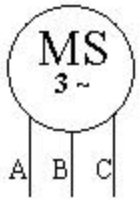


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Input/Output Quantities

• Example

• References

Description

The model represents a synchronous machine with permanent magnet excitation and without damper as a lumped circuit component. Depending on the parameter set, the machine can be operate either as a salient or non-salient pole rotor. The circuit nodes A – B – C are the terminals of star-connected stator winding. To define the parameter values, enter a numerical value, a variable, or expression in the corresponding field of parameter list.

In the case of identical parameters for the inductances L1D and L1Q, a synchronous machine with permanent magnet excitation and non-salient pole rotor is modeled. To model a permanent magnet salient-pole machine, the parameters must be different for L1D and L1Q

The equation system is implemented in a rotor-fixed (rotor-fixed and also rotor flux-fixed) coordinate system (d-q-coordinates). Index 1 represents the stator quantities. The phase quantities of the real three-phase synchronous machine are indicated with a, b, c.

Note: If the line-to-line voltage v_{ab}, v_{bc}, v_{ac} or the line currents i_a, i_b, i_c of the synchronous machine are of special interest, voltmeters or ammeters can be connected to the synchronous machine.

Top

Assumptions and Limitations

Model Limits of Synchronous Machine Models

• 3-phase symmetrical synchronous machine with internal-field system and star-connected stator winding without neutral node (no zero phase-sequence system).

• Linear and iron-loss free magnetic circuit.

• No consideration of skin effects in the windings.

• Exclusive consideration of fundamental flux linkage between stator, damper and excitation windings.

• Rotor position-independent leakage inductances.

• Friction losses (parasitic torques) are not considered in the model; they can be added with the load torque parameter externally.

Top

Mathematical Description

Top

Netlist Syntax

COUPL symp ?InstanceName(@InstanceName):(@(Refbase)@(ID)) a := %0 , b := %1 , c := %2 (l1d := @l1d , l1q := @l1q , ke := @ke , p := @p , j := @j , i1a0 := @i1a0 , i1b0 := @i1b0 , i1c0 := @i1c0 , n0 := @n0 , phi0 := @phi0 , load := @load , r1 := @r1) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
a	Node A	electrical
b	Node B	electrical
c	Node C	electrical

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]

load	Load Torque	real	0.0 [Nm]
r1	Stator Resistance	real	0.4 [Ohm]
l1d	Stator Inductance d-Axis	real	0.042 [H]
l1q	Stator Inductance q-Axis	real	0.042 [H]
ke	Rotor Flux-linkage	real	0.875 [Vs]
p	Number of Pole Pairs	real	2.0 [/]
j	Moment of Inertia	real	0.075 [kg*m^2]
i1a0	Initial Current Stator Phase a	real	0.0 [A]
i1b0	Initial Current Stator Phase b	real	0.0 [A]
i1c0	Initial Current Stator Phase c	real	0.0 [A]
n0	Initial Rotor Speed	real	0.0 [rpm]
phi0	Initial Rotor Position	real	0.0 [rad]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
n	Rotor Speed [rpm]	Output	real
phi	Rotor Angle [rad]	Output	real

Top

Example

This example demonstrates the use of a Permanent Magnetic Synchronous machine as an electrical generator.

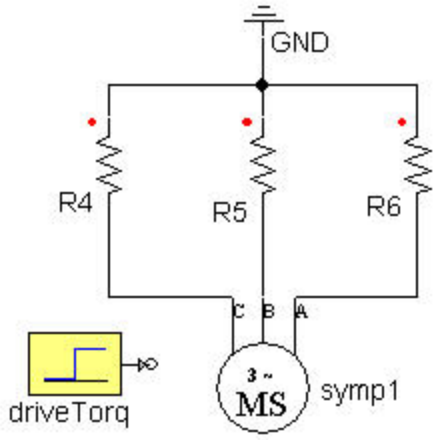


Figure 2. Application examples of the VHDL-AMS Permanent Magnet Synchronous Machine without Damper

Table 4. System Parameters

Component	Parameter	Value [unit]
Step Function driveTorq	T0	0 [S]
	ampl	20
PM Synchronous without Damper symp1	ke	0.85 [H]
	j	0.075
	load	driveTorq.VAL
	r1	0.41 [Ohm]
Resistor r4/r5/r6	r	15 [Ohm]

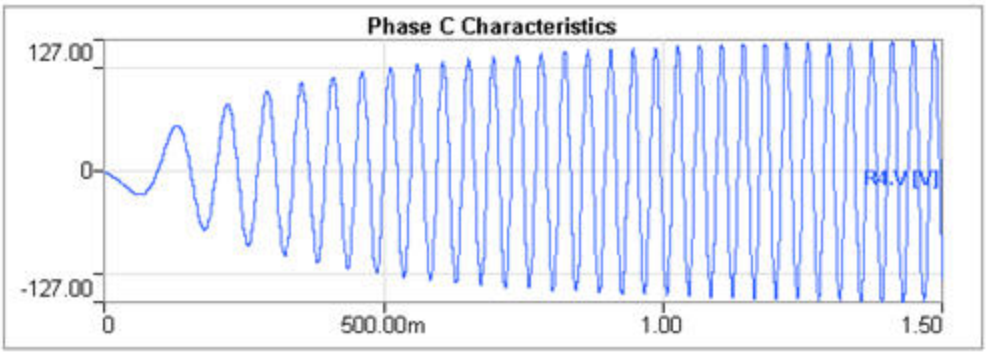


Figure 3. Simulation results-output voltage in Phase C of symp1.

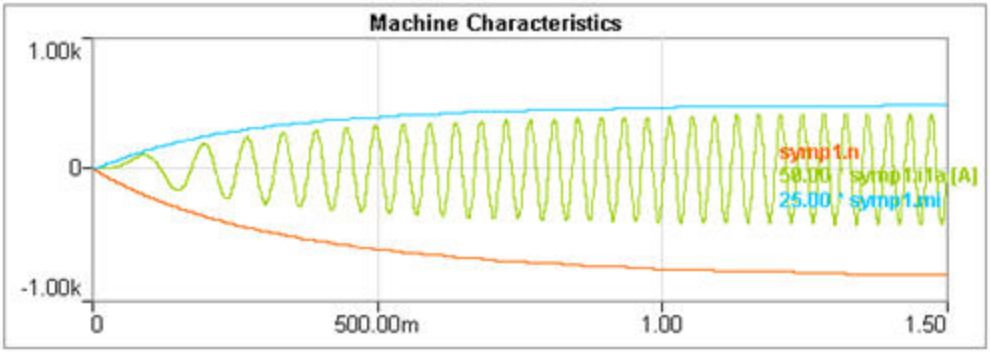


Figure 4. Simulation results-machine speed (symp1.n), phase A current (symp1.i1a), and electromagnetic torque (symp1.mi).

Top

References

Synchronous Machine Permanent Excitation with Damper

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description
• Assumptions and Limitations
• Mathematical Description
• Netlist Syntax
• Conservative Pins
• Parameters
• Input/Output Quantities
• Example
• References

Description

The model represents a permanent excitation synchronous machine with damper circuit i as a lumped circuit component. The damper circuit is not accessible from outside the machine; the corresponding phase windings are connected at a virtual neutral node. Depending on the parameter set, the machine can be operated either as a salient or non-salient pole rotor. The circuit nodes A – B – C are the terminals of star-connected stator winding. To define the parameter values, enter a numerical value, a variable, or expression in the corresponding field of parameter list.

In the case of identical parameters for the inductances L1D and L1Q, a synchronous machine with permanent magnet and non-salient pole rotor is modeled. To model a permanent magnet salient-pole machine, the parameters must be different for L1D and L1Q.

The equation system is implemented in a rotor-fixed (rotor-fixed and also rotor flux-fixed) coordinate system (d-q-coordinates). Index 1 represents the stator quantities; the index 2 the equivalent

quantities of the damper circuit belonged to the rotor. The phase quantities of the real three-phase synchronous machine are indicated with *a*, *b*, *c*.

Note: If the line-to-line voltage v_{ab}, v_{bc}, v_{ac} or the line currents i_a, i_b, i_c of the synchronous machine are of special interest, voltmeters or ammeters can be connected to the synchronous machine.

Top

Assumptions and Limitations

Model Limits of Synchronous Machine Models

• 3-phase symmetrical synchronous machine with internal-field system and star-connected stator winding without neutral node (no zero phase-sequence system).

• Linear and iron-loss free magnetic circuit.

• No consideration of skin effects in the windings.

• Exclusive consideration of fundamental flux linkage between stator, damper and excitation windings.

• Rotor position-independent leakage inductances.

• Friction losses (parasitic torques) are not considered in the model; they can be added with the load torque parameter externally.

Top

Mathematical Description

Top

Netlist Syntax

COUPL sympd ?InstanceName(@InstanceName):(@(Refbase)@(ID)) a := %0 , b := %1 , c := %2 (l1d := @l1d , l1q := @l1q , l2d := @l2d , l2q := @l2q , lm12d := @lm12d , lm12q := @lm12q , ke := @ke , p := @p , j := @j , i1a0 := @i1a0 , i1b0 := @i1b0 , i1c0 := @i1c0 , i2a0 := @i2a0 , i2b0 := @i2b0 , i2c0 := @i2c0 , n0 := @n0 , phi0 := @phi0 , load := @load , r1 := @r1 , r2 := @r2) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:-:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
a	Node A	electrical
b	Node B	electrical
c	Node C	electrical

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
load	Load Torque	real	0.0 [Nm]
r1	Stator Resistance	real	0.4 [Ohm]
r2	Damper Resistance	real	0.24 [Ohm]
l1d	Stator Inductance d-Axis Leakage inductance of stator side	real	0.042 [H]
l1q	Stator Inductance q-Axis Leakage inductance of stator side	real	0.042 [H]
l2d	Damper Inductance d-Axis Leakage inductance of damper circuit side	real	0.042 [H]
l2q	Damper Inductance q-Axis Leakage inductance of damper circuit side	real	0.042 [H]
lm12d	Mutual Inductance stator-damper d-Axis	real	0.041 [H]
lm12q	Mutual Inductance stator-damper q-Axis	real	0.041 [H]
ke	Rotor Flux Linkage	real	0.875 [Vs]
p	Number of Pole Pairs	real	2.0 [/]
j	Moment of Inertia	real	0.075 [kg*m^2]
i1a0	Initial Current Stator Phase a	real	0.0 [A]
i1b0	Initial Current Stator Phase b	real	0.0 [A]

i1c0	Initial Current Stator Phase c	real	0.0 [A]
i2a0	Initial Current Damper Phase a	real	0.0 [A]
i2b0	Initial Current Damper Phase b	real	0.0 [A]
i2c0	Initial Current Damper Phase c	real	0.0 [A]
n0	Initial Rotor Speed	real	0.0 [rpm]
phi0	Initial Rotor Position	real	0.0 [rad]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
n	Rotor Speed [rpm]	Output	real
phi	Rotor Angle [rad]	Output	real

Top

Example

This example demonstrates the use of a Permanent Magnet Synchronous machine with Damper configured as a motor driving a load.

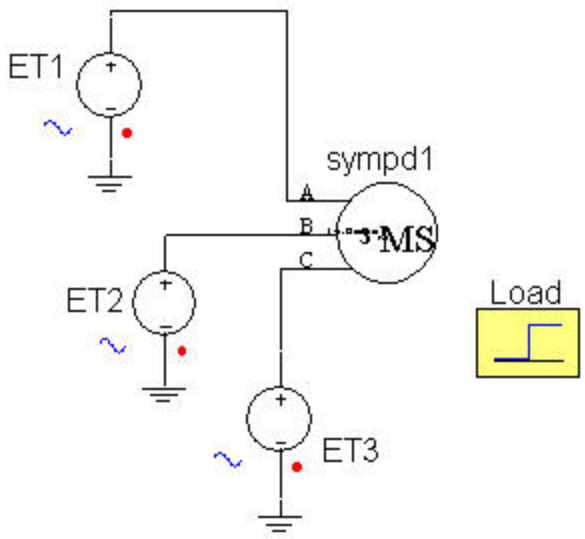


Figure 2. Application examples of the VHDL-AMS Permanent Magnet Synchronous machine with Damper.

Table 4. System Parameters

Component	Parameter	Value [unit]
Time Controlled Voltage Source (Sinusoidal) ET1/ET2/ET3	freq	50 [Hz]
	tperio	20m [S]
	ampl	0.326k [V]
	phase ET1/ET2/ET3	0/-120/-240 [Deg]
Step Function Load	T0	0.2 [S]
	ampl	60
PM Synchronous with Damper sympd1	p	2
	j	0.075
	load	Load.VAL
	r1	0.4 [Ohm]
	r2	0.24 [Ohm]

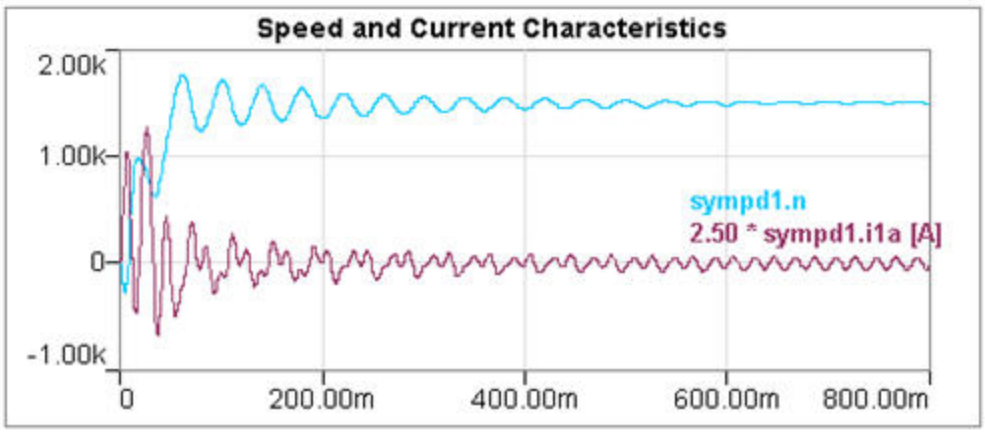


Figure 3. Simulation results-speed and phase A current of sympd1.

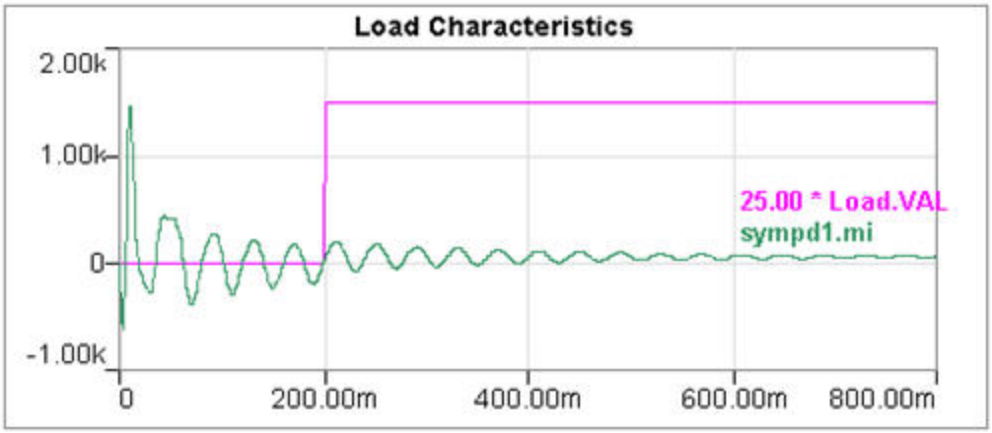


Figure 4. Simulation results-load and electromagnetic torque of sympd1.

Top

References

Passive Elements

- Capacitor in VHDL-AMS (c)
- Conductor in VHDL-AMS (g)
- Inductor in VHDL-AMS (l)
- Resistor in VHDL-AMS (r)

Capacitor

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

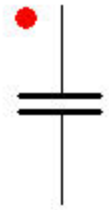


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents an electrical capacitance. To define the capacitance value and the initial voltage (static value), enter a numerical value, a variable, or expression in the parameter list. A flag can be set to specify whether or not the initial voltage is used.

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

COUPL c ?InstanceName(@InstanceName):(@(Refbase)@(ID)) p := %0 , m := %1 (v0 := @v0 , use_v0 := @use_v0 , c := @c) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
p	Plus Terminal (with red point)	electrical
m	Minus Terminal	electrical

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
c	Capacitance	real	1.0e-6 [F]
v0	Initial Voltage	real	0.0 [V]
use_v0	If set to 1 (true), the initial condition specified in v0 will be used. If set to (0) false, the model assumes that conditions have reached steady state - that is I=0, and current will be determined based on the rest of the circuit.	Boolean	0 (false)

Top

Example

This example is shows a linear Capacitance being charged by a voltage source through a linear resistor. In the example, two different ways of specifying the capacitance value are demonstrated, a simple linear case as shown in Figure 2, and an expression controlled case as shown in Figure 4.

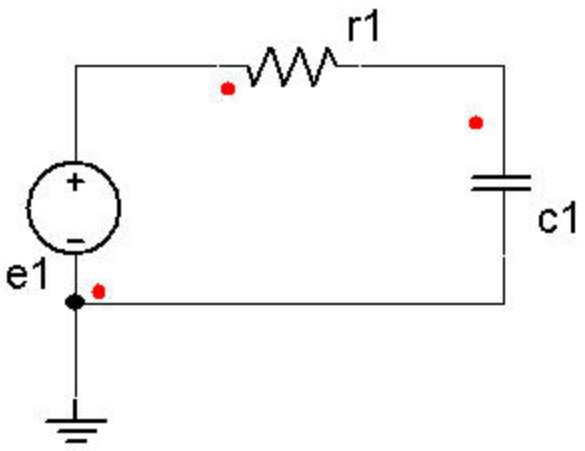


Figure 2. Application examples of the VHDL-AMS Linear Capacitance model

Table 3. System Parameters

Component	Parameter	Value [unit]
Voltage Source e1	emf	100 [V]
Capacitor c1	c	1u [F]
	v0	0 [V]
	use_v0	1 (true)
Resistor r1	r	10k [Ohm]

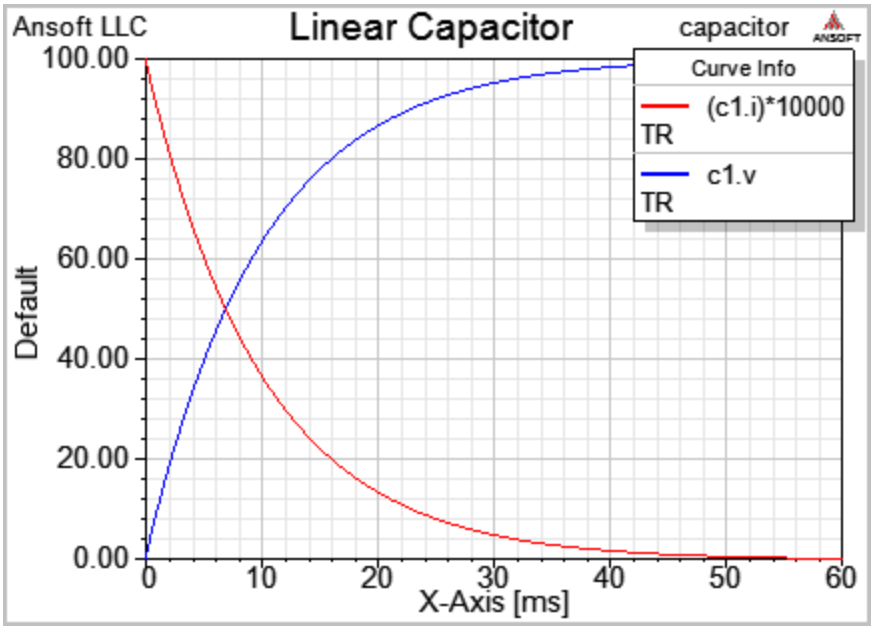


Figure 3. Simulation results-voltage across and current through capacitance c1

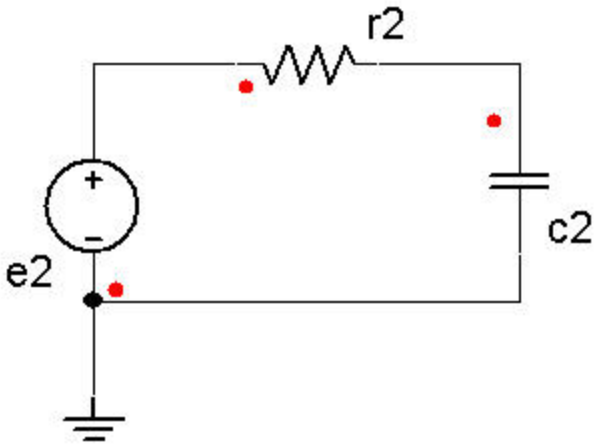


Figure 4. Application examples of the VHDL-AMS Linear Capacitance model with expression controlled capacitance value

Table 4. System Parameters

Component	Parameter	Value [unit]
Voltage Source e2	emf	10 [V]
Capacitor c2	c	0.1*t+0.005 [F]
	v0	0 [V]

	use_v0	1 (true)
Resistor r2	r	100 [Ohm]

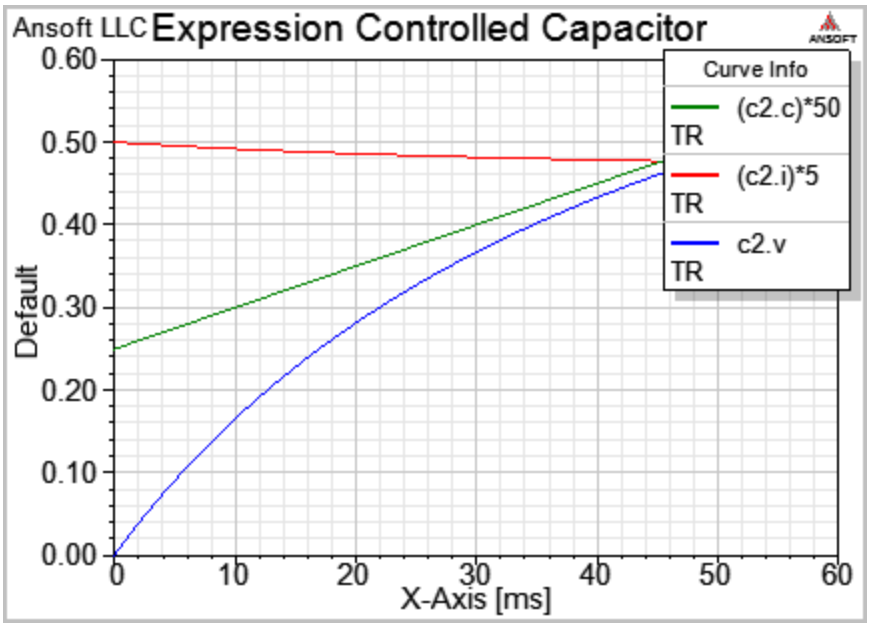


Figure 5. Simulation results-voltage across, current through, and capacitance value of c2

Top

References

Conductor

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents an electrical conductance. To define the conductance value, enter a numerical value, a variable, or expression in the parameter list.

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

COUPL g ?InstanceName(@InstanceName):(@(Refbase)@(ID)) p := %0 , m := %1 (g := @g) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:-:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
p	Plus Terminal (with red point)	electrical
m	Minus Terminal	electrical

Top

Parameters

Table 2

Name	Description	Data Type	Default Value[Unit]
g	Conductivity	real	1.0e-3 [S]

Top

Example

This example is comprised of a linear Conductance, in series with a capacitor, driven by a voltage source. In the example, two different ways of specifying the conductance value are demonstrated, a simple linear case as shown in Figure 2, and an expression controlled case as shown in Figure 4.

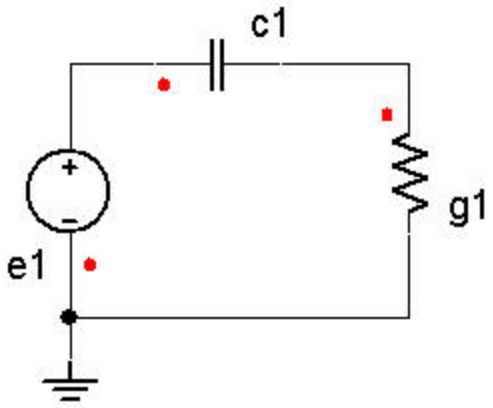


Figure 2. Application examples of the VHDL-AMS Linear Conductance model

Table 3. System Parameters

Component	Parameter	Value [unit]
Voltage Source e1	emf	1 [V]

Capacitance c1	c	1 [uF]
	v0	0 [V]
Conductance g1	g	1m [S]

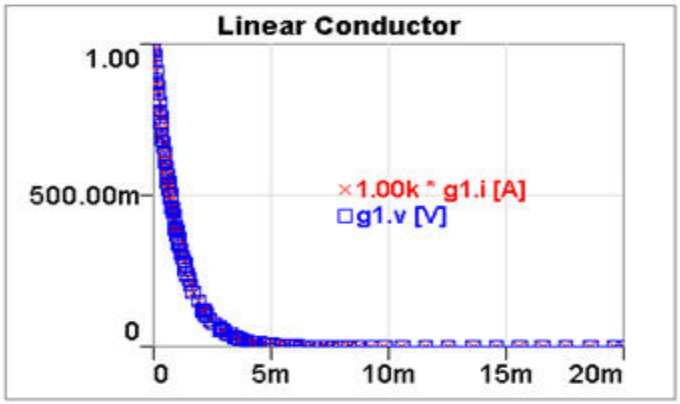


Figure 3. Simulation results-voltage across and current through conductance g1

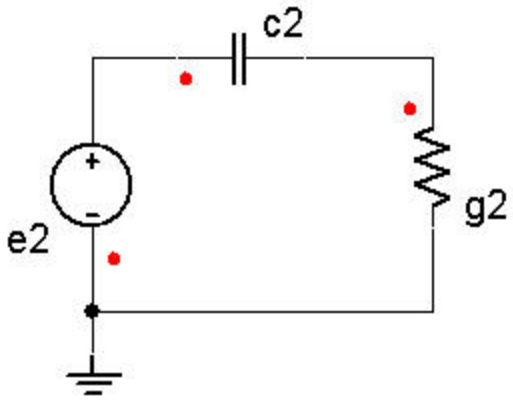


Figure 4. Application examples of the VHDL-AMS Linear Conductance model with expression controlled value

Table 4. System Parameters

Component	Parameter	Value [unit]
Voltage Source e2	emf	1 [V]
Capacitance c2	c	1 [uF]

	v0	0 [V]
Conductance g2	g	2*t+0.005 [S]

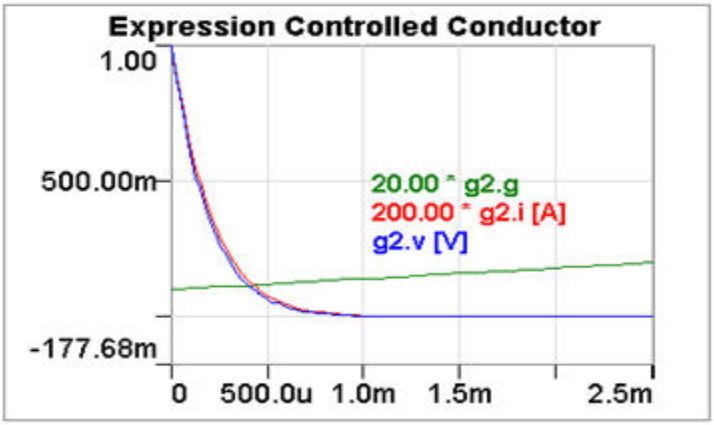


Figure 5. Simulation results-voltage across, current through, and conductance value of g2

Top

References

Inductor

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents an electrical inductance. To define the inductance value and the initial current (static value), enter a numerical value, a variable, or expression in the parameter list. A flag can be set to specify whether or not the initial current is used.

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

COUPL I ?InstanceName(@InstanceName):(@(Refbase)@(ID)) p := %0 , m := %1 (i0 := @i0 , use_i0 := @use_i0 , I := @I) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
p	Plus Terminal (with red point)	electrical
m	Minus Terminal	electrical

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
l	Inductance	real	1.0e-3 [H]
i0	Initial Current	real	0.0 [A]
use_i0	If set to 1 (true), the initial condition specified in i0 will be used. If set to (0) false, the model assumes that conditions have reached steady state - that is V=0, and voltage will be determined based on the rest of the circuit.	Boolean	0 (false)

Top

Example

This example shows a linear Inductor being driven by a current source through a linear resistor. In the example, two different ways of specifying the inductance value are demonstrated, a simple linear case as shown in Figure 2, and an expression controlled case as shown in Figure 4.

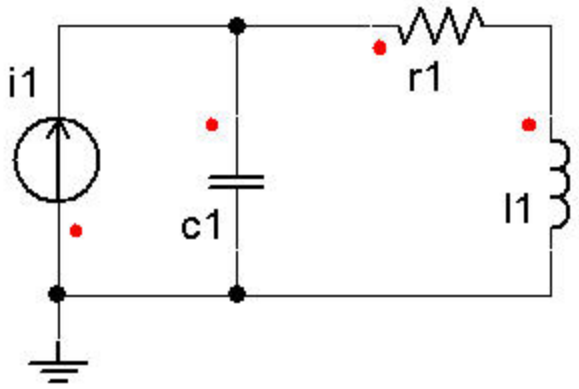


Figure 2. Application examples of the VHDL-AMS Linear Inductor model

Table 3. System Parameters

Component	Parameter	Value [unit]
Current Source i1	i	1 [A]
Capacitor c1	c	1u [F]
	v0	0 [V]
	use_v0	1 (true)
Resistor r1	r	1k [Ohm]
Inductor l1	l	1m [H]
	i0	0 [A]
	use_i0	1 (true)

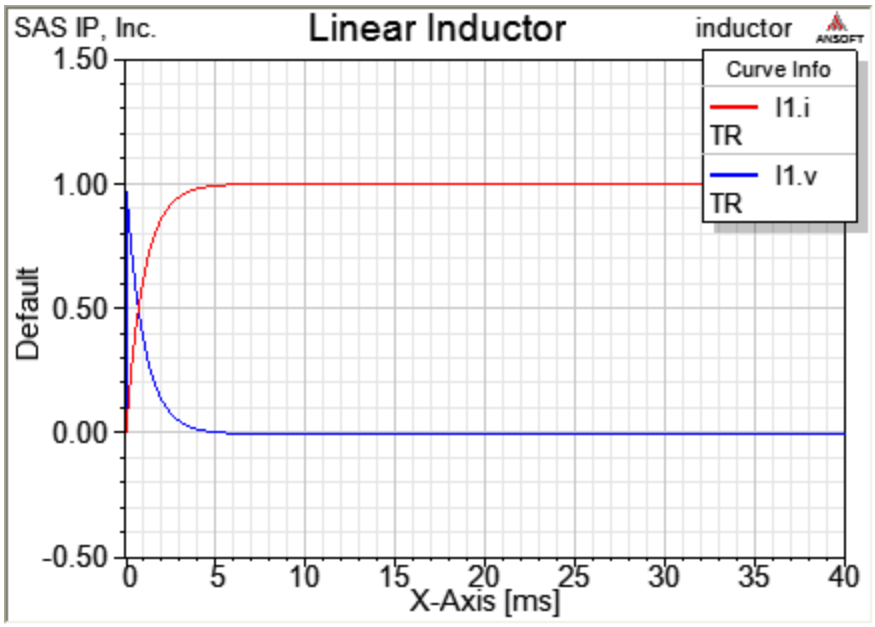


Figure 3. Simulation results-voltage across and current through inductor I1

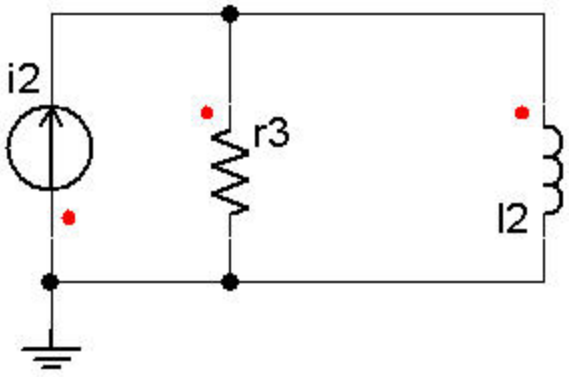


Figure 4. Application examples of the VHDL-AMS Linear Inductor model with expression controlled inductance value

Table 4. System Parameters

Component	Parameter	Value [unit]
Current Source i2	i	1 [A]
Inductor I2	I	5*t+0.5 [H]
	i0	0 [A]

	use_i0	1 (true)
Resistor r2	r	100 [Ohm]

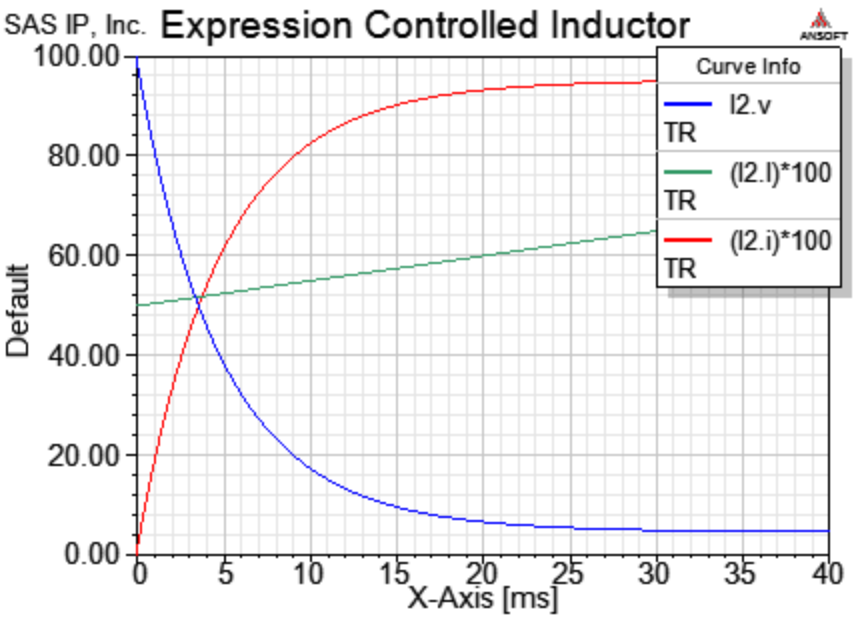


Figure 5. Simulation results-voltage across, current through, and inductance value of L2

Top

References

Resistor

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents an electrical resistance. To define the resistance value, enter a numerical value, a variable, or expression in the parameter list.

To avoid initialization problems with the simulator, do not use a block input signal as parameter value.

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

COUPL r ?InstanceName(@InstanceName):(@(Refbase)@(ID)) p := %0 , m := %1 (r := @r) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:-:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
p	Plus Terminal (with red point)	electrical
m	Minus Terminal	electrical

Top

Parameters

Table 2

Name	Description	Data Type	Default Value[Unit]
r	Resistance	real	1.0e3 [Ohm]

Top

Example

This example is comprised of a simple linear resistor driven by a current source. The value of the resistor may be a single value or controlled by a variable or an expression. In this example, two cases are demonstrated, the simple resistor value as shown in Figure 2. and the expression controlled case as shown in Figure 4.

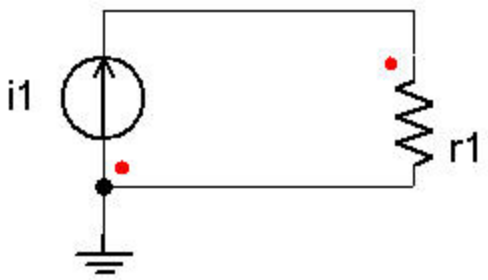


Figure 2. Application examples of the VHDL-AMS Linear Resistor model

Table 3. System Parameters

Component	Parameter	Value [unit]
Current Source i1	i	1 [A]
Resistor (Linear) r1	r	2 [Ohm]

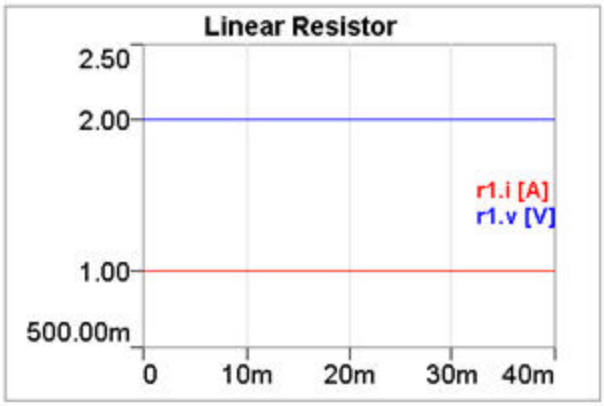


Figure 3. Simulation results-voltage across and current through r1

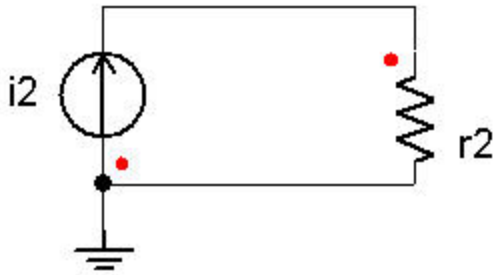


Figure 4. Application example of the VHDL-AMS Linear Resistor model with expression controlled resistance

Table 4. System Parameters for Expression Controlled Resistor

Component	Parameter	Value [unit]
Current Source i2	i	1 [A]
Resistor r2	r	10*t+2 [Ohm]

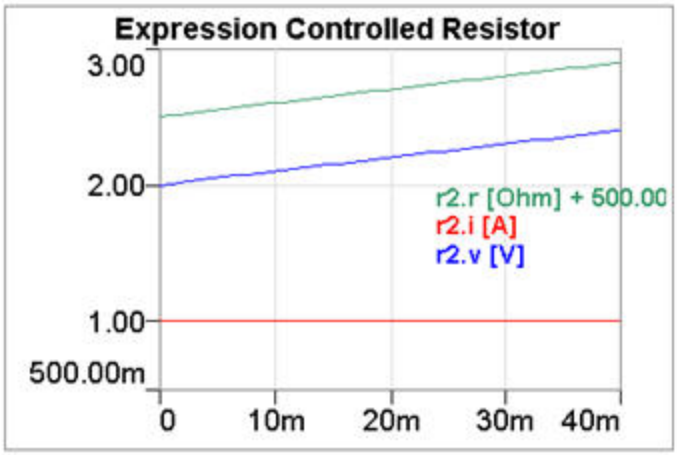


Figure 5. Simulation results-voltage across, current through, and resistance versus time of r2

Top

References

Semiconductors System Level

- BJT Model in VHDL-AMS (bjt)
- Diode in VHDL-AMS (d)
- IGBT Model in VHDL-AMS (igbt)
- MOSFET Model in VHDL-AMS (mos)

Bipolar Junction Transistor

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

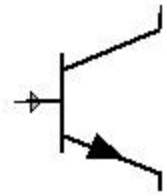


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The System level BJT is used to simulate a static voltage-current-relation. Each voltage cause a corresponding current depending on the applied control signal.

To control the BJT, a logical signal is applied. This signal can originate from any quantity used in the simulation model.

The BJT characteristic is defined by an exponential function. You have to enter values for the saturation current, thermal voltage, and reverse resistance.

Control Signal (Base current)	Component State
> 0	On
<= 0	Off

The BJT model has two architectures that describe two different behaviors. The "behav" architecture describes the BJT characteristics with an exponential function and parameter values for the saturation current, thermal voltage and reverse resistance have to be provided. The "equiv" architecture describes the BJT characteristics with an equivalent line and the forward voltage, reverse resistance, and bulk resistance have to be provided. The required architecture can be chosen in the model instance's properties dialog from the library tab.

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

COUPL bjt ?InstanceName(@InstanceName):(@(Refbase)@(ID)) c := %0 , e := %1 (isat := @isat , vt := @vt , rr := @rr , vf := @vf , rb := @rb , ctrl := @ctrl) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
C	Collector	electrical
E	Emitter	electrical

Top

Parameters

Table 2

Name	Description	Data Type	Default Value[Unit]
rr	Reverse Resistance	real	100e3 [Ohm]
isat	Saturation Current	real	1.0e-12 [A]
vf	Forward Voltage	real	0.8 [V]
ctrl	Control Signal	real	0.0
vt	Threshold Voltage	real	35.0e-3 [V]
rb	Bulk Resistance	real	1.0e-3 [Ohm]

Top

Example

This example demonstrates the use of a Bipolar Junction Transistor Model to switch a DC source to a load resistor. The BJT ctrl input is controlled by a triangle wave source that switches the device into an on or off state.

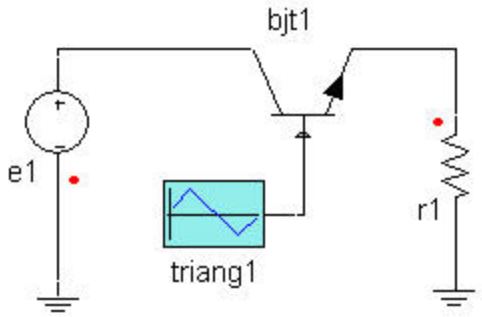


Figure 2. Application examples of the VHDL-AMS BJT Transistor model

Table 3. System Parameters

Component	Parameter	Value [unit]
Voltage Source e1	emf	300 [V]
Triangular Wave Source triang1	tdelay	0 [S]
	ampl	311 [V]
	off	0 [S]
	freq	300 [Hz]
BJT bjt1	isat	1p [A]
	vt	35m [V]
	rr	100k [Ohm]
	vf	0.8 [V]
	rb	1m [Ohm]
	ctrl	triang1.val
Resistor r1	r	1k [Ohm]

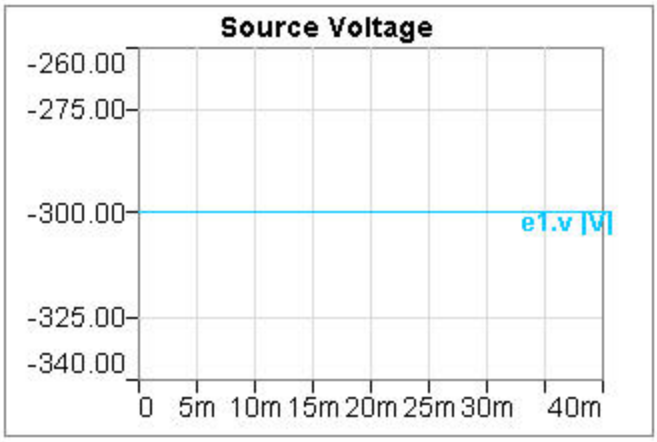


Figure 3. Simulation results-DC input from source e1.

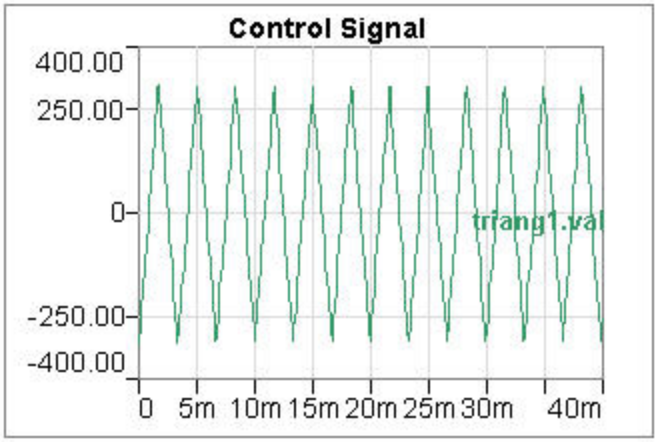


Figure 4. Simulation results-control signal to ctrl input of BJT bjt1.

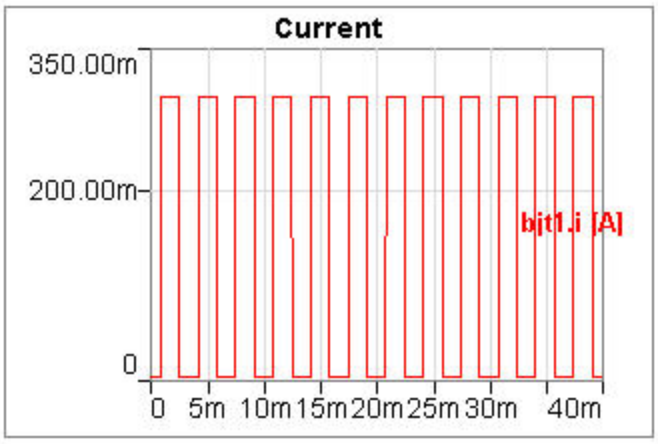


Figure 5. Simulation results-voltage output of BJT bjt1.

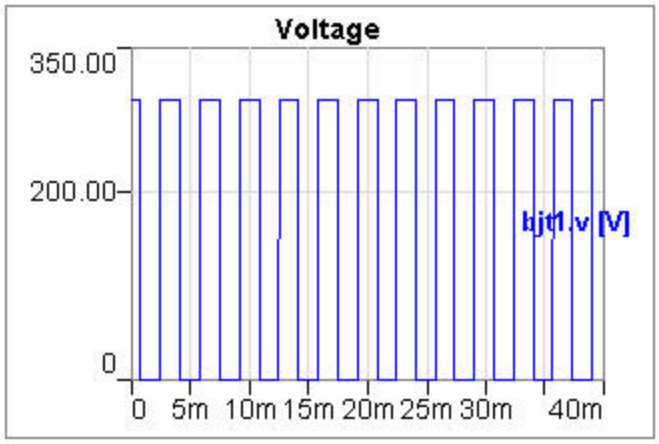


Figure 6. Simulation results-current output of BJT bjt1.

Top

References

Diode

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The System level Diode model is used to simulate a static voltage-current-relation. Each voltage cause a corresponding current.

The diode characteristic is defined by an exponential function. You have to enter values for the saturation current, thermal voltage, and reverse resistance.

The diode model has two architectures that describe two different behaviors. The "behav" architecture describes the diode characteristics with an exponential function and parameter values for the saturation current, thermal voltage and reverse resistance have to be provided. The "equiv" architecture describes the diode characteristics with an equivalent line and the forward voltage, reverse resistance, and bulk resistance have to be provided. The required architecture can be chosen in the model instance's properties dialog from the library tab.

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

COUPL d ?InstanceName(@InstanceName):(@(Refbase)@(ID)) p := %0 , m := %1 (isat := @isat , vt := @vt , rr := @rr , vf := @vf , rb := @rb) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
p	Plus terminal	electrical
m	Minus Terminal	electrical

Top

Parameters

Table 2

Name	Description	Data Type	Default Value[Unit]
rr	Reverse Resistance	real	100.0e3 [Ohm]
isat	Saturation Current	real	1.0e-12 [A]
vt	Threshold Voltage	real	35.0e-3 [V]
rb	Bulk Resistance	real	1.0e-3 [Ohm]
vf	Forward Voltage	real	0.8 [V]

Top

Example

This example demonstrates the use of a Semiconductor System Diode Model to clip a sinusoidal waveform to a load resistor.

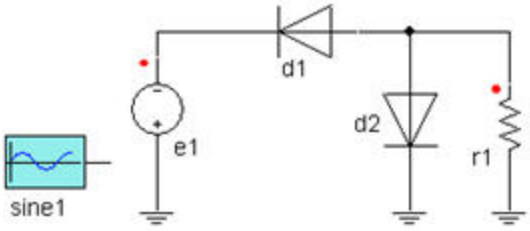


Figure 2. Application examples of the VHDL-AMS Diode model

Table 3. System Parameters

Component	Parameter	Value [unit]
Voltage Source e1	emf	sine1.val
Diode d1/d2	isat	1p [A]
	vt	35m [V]
	vf	0.8 [V]
	rr	100k [Ohm]
	rb	1m [Ohm]
Sine Wave Source sine1	tdelay	0 [S]
	ampl	326 [V]
	freq	50 [Hz]
Resistor r1	r	1k [Ohm]

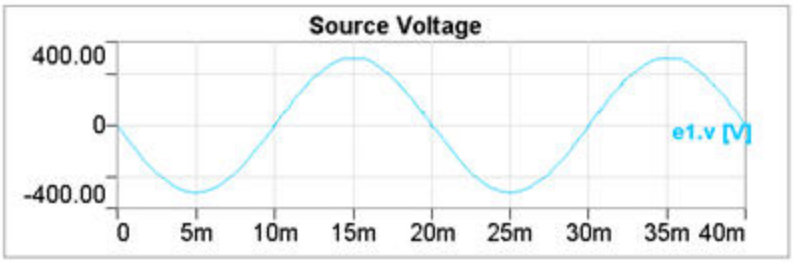


Figure 3. Simulation results-output voltage of the voltage source e1.

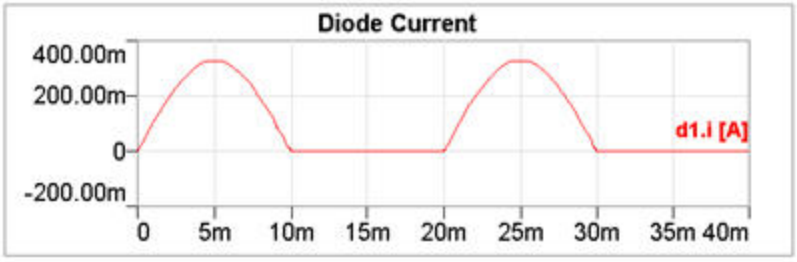


Figure 4. Simulation results-current through diode d1.

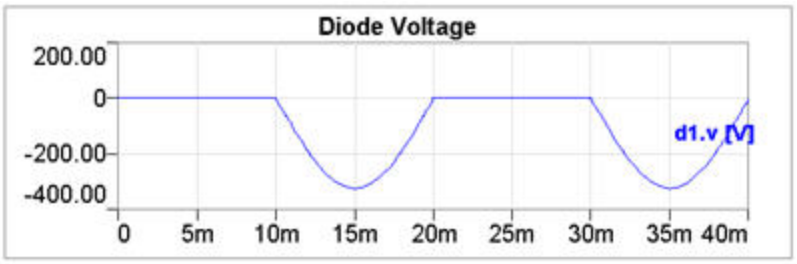


Figure 5. Simulation results-voltage across diode d1.

Top

References

IGBT

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

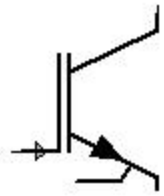


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The System level IGBT (Insulated Gate Bipolar Transistor) model is used to simulate a static voltage-current-relation. Each voltage cause a corresponding current depending on the applied control signal.

To control the IGBT, a logical signal is applied. This signal can originate from any quantity used in the simulation model.

The IGBT characteristic is defined by an exponential function. You have to enter values for the saturation current, thermal voltage, and reverse resistance.

Control Signal (Base current)	Component State
> 0	On
<= 0	Off

The IGBT model has two architectures that describe two different behaviors. The "behav" architecture describes the IGBT characteristics with an exponential function and parameter values for the saturation current, thermal voltage and reverse resistance have to be provided. The "equiv" architecture describes the IGBT characteristics with an equivalent line and the forward voltage,

reverse resistance, and bulk resistance have to be provided. The required architecture can be chosen in the model instance's properties dialog from the library tab.

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

COUPL igbt ?InstanceName(@InstanceName):(@(Refbase)@(ID)) c := %0 , e := %1 (isat := @isat , vt := @vt , rr := @rr , vf := @vf , rb := @rb , ctrl := @ctrl) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
c	Collector	electrical
e	Emitter	electrical

Top

Parameters

Table 2

Name	Description	Data Type	Default Value[Unit]
rr	Reverse Resistance	real	100.0e3 [Ohm]
isat	Saturation Current	real	1.0e-12 [A]
vf	Forward Voltage	real	0.8 [V]
ctrl	Control Signal	real	0.0
vt	Threshold Voltage	real	35.0e-3 [V]
rb	Bulk Resistance	real	1.0e-3 [Ohm]

Top

Example

This example demonstrates the use of an IGBT Model in a switch-mode circuit. The IGBT ctrl input is controlled by a triangle wave source.

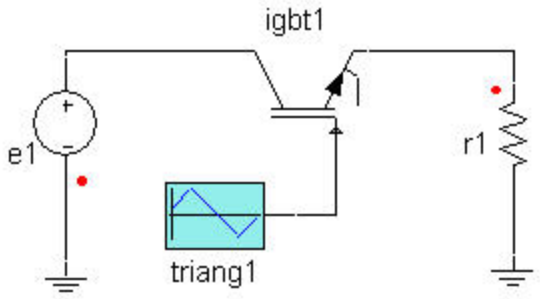


Figure 2. Application examples of the VHDL-AMS IGBT model

Table 3. System Parameters

Component	Parameter	Value [unit]
Voltage Source e1	emf	300 [V]
IGBT igbt1	isat	1p [A]
	vt	100m [V]
	rr	100 [Ohm]
	vf	0.8 [V]
	rb	1m [Ohm]
	ctrl	triang1.val
Triangular Wave Source triang1	tdelay	0 [S]
	ampl	311 [V]
	off	0 [S]
	freq	300 [Hz]

Resistor r1	r	1k [Ohm]

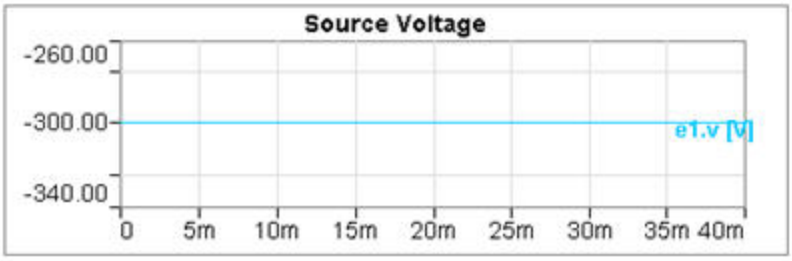


Figure 3. Simulation results-voltage output of voltage source e1.

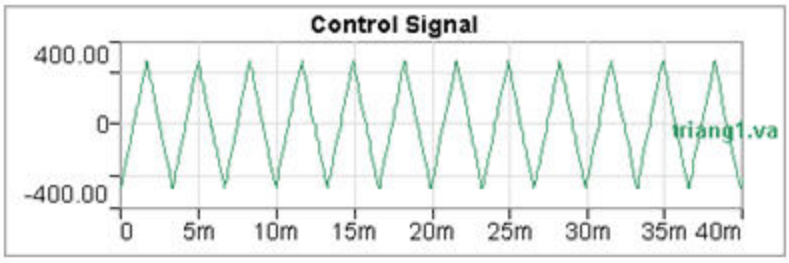


Figure 4. Simulation results-control signal from Triangular Wave source to IGBT ctrl.

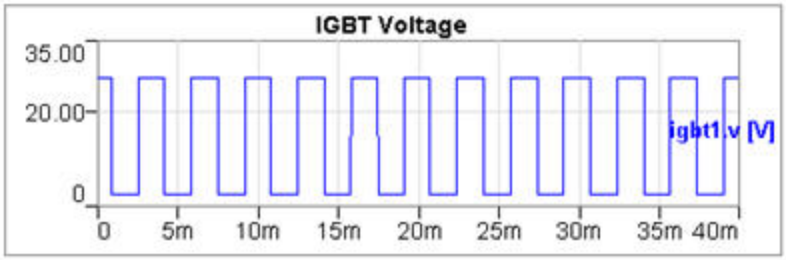


Figure 5. Simulation results-output voltage of igbt1.

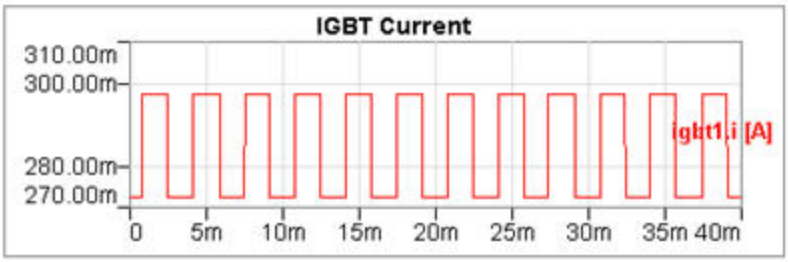


Figure 6. Simulation results-output current of igbt1.

Top

References

MOS Fieldeffect Transistor

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

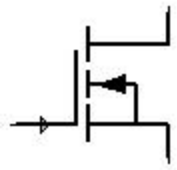


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The System level MOS model is used to simulate a static voltage-current-relation. Each voltage cause a corresponding current depending on the applied control signal.

To control the MOS, a logical signal is applied. This signal can originate from any quantity used in the simulation model.

The MOS characteristic is defined by an exponential function. You have to enter values for the saturation current, thermal voltage, and reverse resistance.

Control Signal (Base current)	Component State
> 0	On
<= 0	Off

The MOSFET model has two architectures that describe two different behaviors. The "behav" architecture describes the MOSFET characteristics with an exponential function and parameter values for the saturation current, thermal voltage and reverse resistance have to be provided. The "equiv" architecture describes the MOSFET characteristics with an equivalent line and the forward voltage, reverse resistance, and bulk resistance have to be provided. The required architecture can be chosen in the model instance's properties dialog from the library tab.

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

COUPL mos ?InstanceName(@InstanceName):(@(Refbase)@(ID)) d := %0 , s := %1 (isat := @isat , vt := @vt , rr := @rr , vf := @vf , rb := @rb , ctrl := @ctrl) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
d	Drain	electrical
s	Source	electrical

Top

Parameters

Table 2

Name	Description	Data Type	Default Value[Unit]
rr	Reverse Resistance	real	100.0e3 [Ohm]
isat	Saturation Current	real	1.0e-12 [A]
vf	Forward Voltage	real	0.8 [V]
ctrl	Control Signal	real	0.0
vt	Threshold Voltage	real	35.0e-3 [V]
rb	Bulk Resistance	real	1.0e-3 [Ohm]

Top

Example

This example demonstrates the use of a MOSFET Model to switch a DC source to a load resistor. The ctrl input of the MOSFET mos1 is controlled by a triangle wave source that switches

the device into an on or off state.

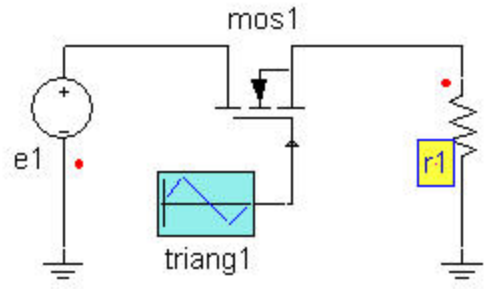


Figure 2. Application examples of the VHDL-AMS MOS Field Effect Transistor model

Table 3. System Parameters

Component	Parameter	Value [unit]
Voltage Source e1	emf	300 [V]
Triangular Wave Source triang1	tdelay	0 [S]
	ampl	311 [V]
	off	0 [S]
	freq	300 [Hz]
MOSfet mos1	isat	1p [A]
	vt	100m [V]
	rr	100 [Ohm]
	vf	0.8 [V]
	rb	1m [Ohm]
	ctrl	triang1.val
Resistor r1	r	1k [Ohm]

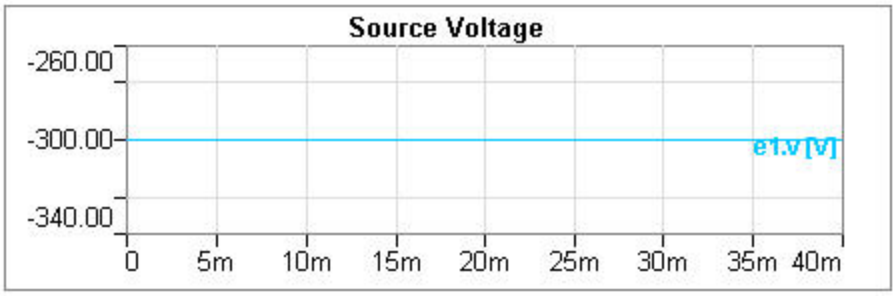


Figure 3. Simulation results-DC input from source (e1).

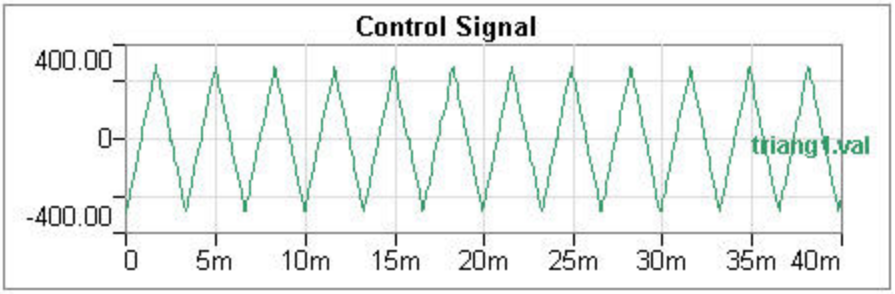


Figure 4. Simulation results-control signal to ctrl input of MOSFET mos1.

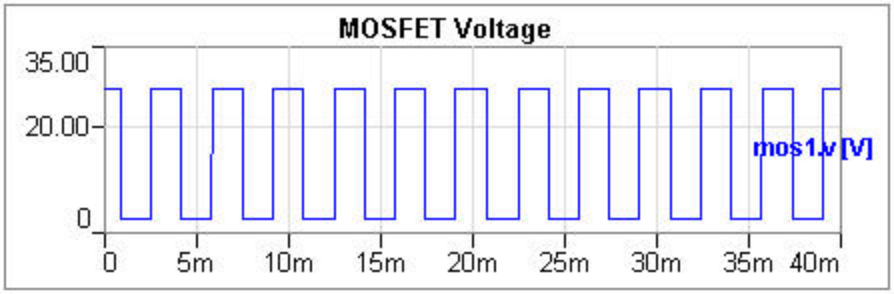


Figure 5. Simulation results-voltage output of MOSFET mos1.

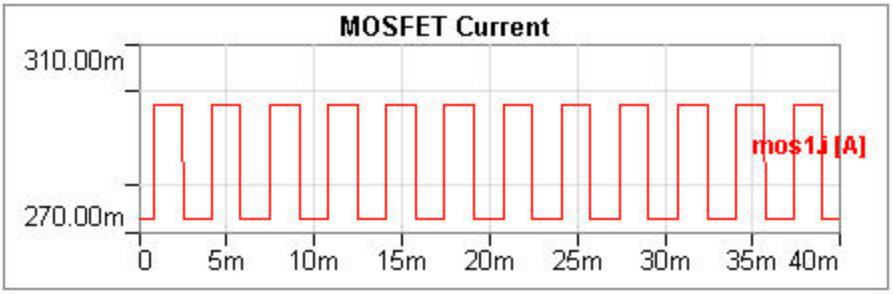


Figure 6. Simulation results-current output of MOSFET mos1.

Top

References

Sources

- Voltage Source in VHDL-AMS (e)
- Controlled Voltage Source in VHDL-AMS (ec)
- Voltage Controlled Oscillator Voltage Source in VHDL-AMS (evco)
- Current Source in VHDL-AMS (i)
- Controlled Current Source (ic)
- Voltage Controlled Oscillator Current Source in VHDL-AMS (ivco)

Voltage Source

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

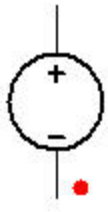


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents an independent voltage source. To define the EMF value, enter a numerical value, a variable, or expression in the parameter list.

The current through an ideal voltage source can be arbitrary. Real voltage sources have always an internal resistance. If the internal resistance is essential for the simulation target, a corresponding resistance must be connected in series.

The Twin Builder voltage sources have the opposite orientation compared to SPICE-like voltage sources.

See also Twin Builder Reference Arrow System

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

COUPL e ?InstanceName(@InstanceName):(@(Refbase)@(ID)) p := %0 , m := %1 (emf := @emf , ac_mag := @ac_mag , ac_phase := @ac_phase) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
p	Plus Terminal (with red point)	electrical
m	Minus Terminal	electrical

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
emf	EMF Value	real	1.0 [V]
ac_mag	Magnitude of EMF (AC)	real	1.0e-3 [V]
ac_ phase	Phase Shift of EMF (AC)	real	0.0 [deg]

Top

Example

This example shows a voltage source driving a linear resistor. In the example, two different ways of specifying the voltage value are demonstrated, a simple constant voltage case as shown in Figure 2, and an expression controlled case as shown in Figure 4.

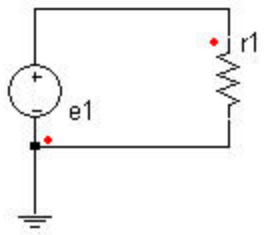


Figure 2. Application examples of the VHDL-AMS Voltage Source model

Table 3. System Parameters

Component	Parameter	Value [unit]
Voltage Source e1	emf	90 [V]

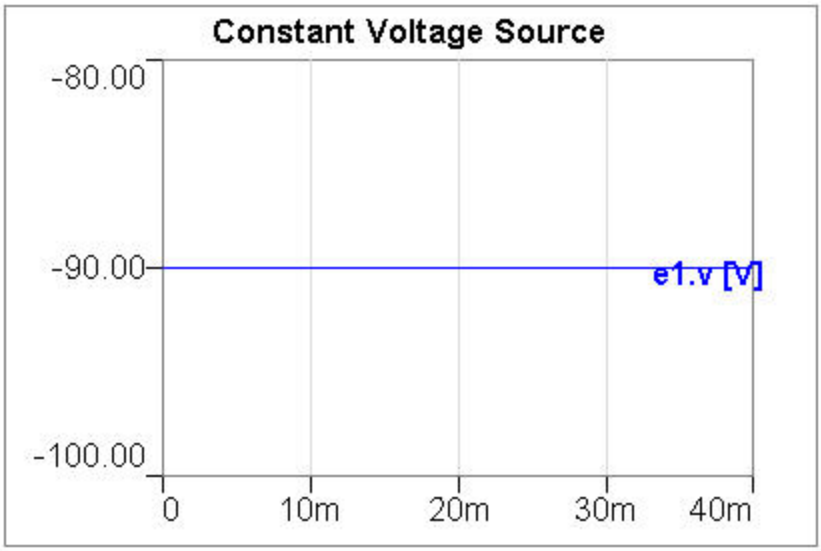


Figure 3. Simulation results-voltage output of voltage source e1.

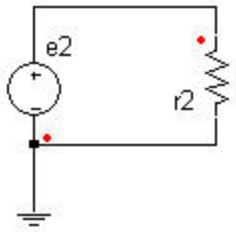


Figure 4. Application examples of the VHDL-AMS Voltage Source model with expression controlled output value

Table 4. System Parameters

Component	Parameter	Value [unit]
Voltage Source e2	emf	10*t [V]

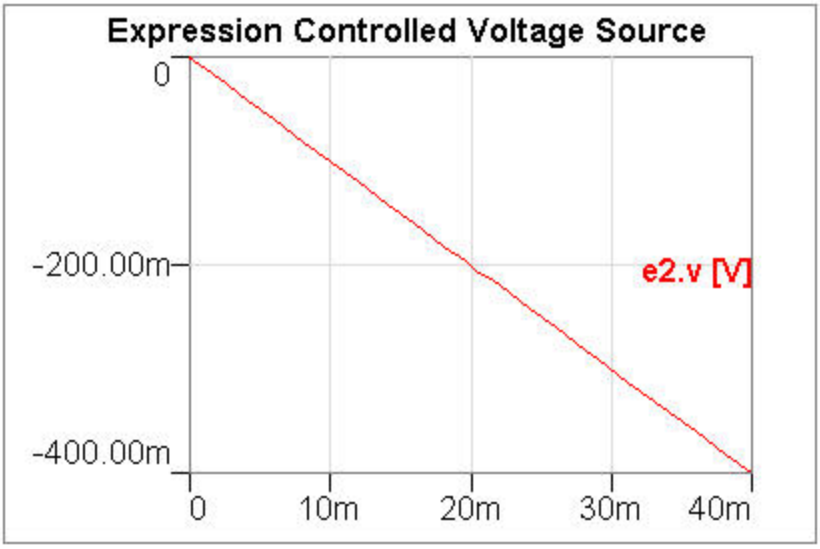


Figure 5. Simulation results-voltage output of voltage source e2 as a function of time.

Top

References

Controlled Voltage Source

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

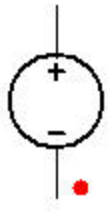


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents a dependent voltage source. The value of the source is calculated from the controlling value multiplied by an arbitrary control coefficient.

To define controlling value and control coefficient, enter a numerical value, a variable, or expression in the parameter list.

See also Twin Builder Reference Arrow System.

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

COUPL ec ?InstanceName(@InstanceName):(@(Refbase)@(ID)) p := %0 , m := %1 (fact := @fact , quant := @quant) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
p	Plus Terminal (with red point)	electrical
m	Minus Terminal	electrical

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
quant	Control Quantity	real	0.0 [V/A]
fact	Factor	real	1.0 [/]
ac_mag	Magnitude of EMF (AC)	real	1.0e-3
ac_ phase	Phase Shift of EMF (AC)	real	0.0

Top

Example

This example demonstrates the use of both current and voltage control of a voltage source. In the example, an expression controlled voltage source (e3) is used to drive a resistor (r3). An ammeter and a voltage meter generate the control signals to drive the controlled sources (ec1 and ec2). The voltage generated by the controlled voltage sources are shown in Figure 3 and Figure 4.

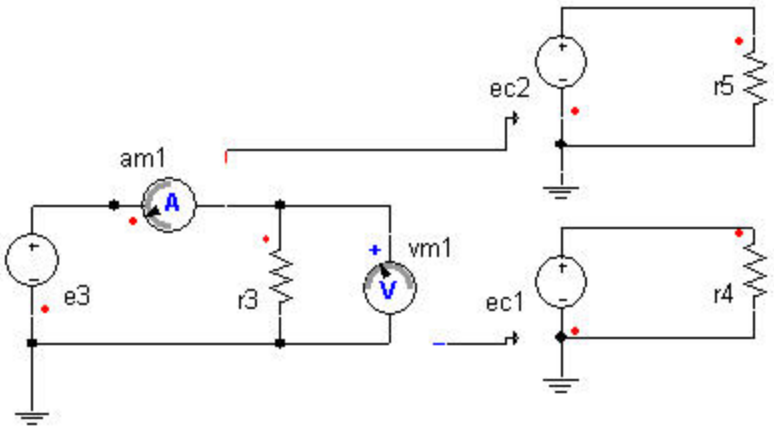


Figure 2. Application examples of the VHDL-AMS Controlled Voltage Source model

Table 3. System Parameters

Component	Parameter	Value [unit]
Voltage Source e3	emf	10*t [V]
Controlled Voltage Source ec1	fact	3
	quant	vm1.v
Controlled Voltage Source ec2	fact	2
	quant	am1.i

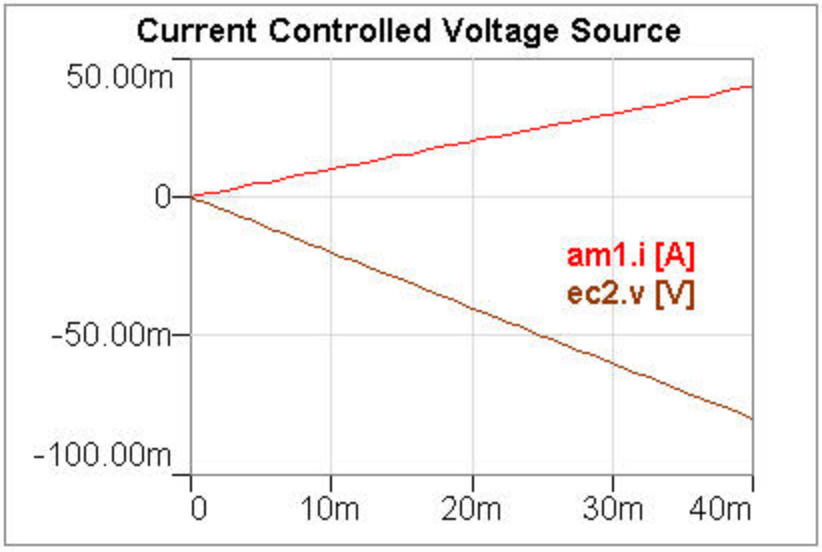


Figure 3. Simulation results-control current at ammeter (am1) and output of controlled voltage source (ec2).

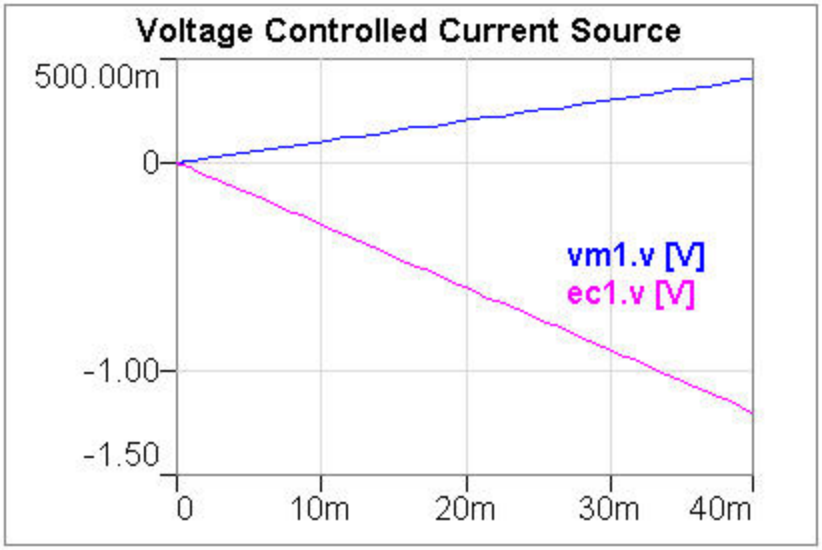


Figure 4. Simulation results-control voltage at voltmeter (vm1) and output of controlled voltage source (ec1).

Top

References

Voltage Controlled Oscillator Voltage Source

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

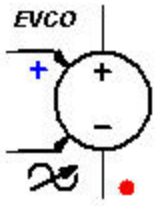


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents a voltage controlled Oscillator (voltage source). The VCO provides a sine wave with a frequency dependent on the input signal. The frequency is changed in relation 1:1, e.g. vcontrol=50V -> f=50Hz. Furthermore you can define amplitude and phase shift of the output sine wave and limit of the control signal.

The limit input voltage value, lmt interprets an input of 1 to indicate that the input voltage is limited and an input of 0 to indicate that the input voltage is not limited. If the lmt field is specified with a value of '1' the entry for the upper limit and lower limit is used for limiting the input voltage. This is represented as a static real value.

The upper limit (ul) and lower limit (ll) values are represented as static real values and used to limit the input voltage if lmt has a value of 1.0.

The amplitude value, ampl defines the amplitude of the output sine wave and is represented as a static real value.

The phase value, phase, specifies the phase shift of the output sine wave. The phase value needs to be specified in radians and is represented as a static real value.

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

COUPL evco ?InstanceName(@InstanceName):(@(Refbase)@(ID)) p := %0 , m := %1 , fp := %2 , fm := %3 (phase := @phase , lmt := @lmt , ul := @ul , ll := @ll , ampl := @ampl , ac_mag := @ac_mag , ac_phase := @ac_phase) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
p	Plus Terminal	electrical
m	Minus Terminal	electrical
fp	Frequency Plus Terminal	electrical
fm	Frequency Minus Terminal	electrical

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
ul	Input Voltage Upper Limit	real	0.0 [V]
ll	Input Voltage Lower Limit	real	0.0 [V]
ampl	Amplitude	real	326.0 [V]
phase	Phase	real	0.0 [rad]
lmt	Limit Input Voltage Flag, 1=yes, 0=no	real	0.0
ac_mag	Magnitude of EMF (AC)	real	1.0e-3 [V]
ac_ phase	Phase Shift of EMF(AC)	real	0.0 [deg]

Top

Example

This example demonstrates the use of a VCO Voltage Source. A Voltage Controlled Oscillator (VCO) provides a sine wave output signal whose frequency depends upon the amplitude of the control signal, in this case a voltage input. In this example a controlled voltage source is driven by a trapezoidal source to provide the varying input to the VCO. The VCO output is a time varying voltage source.

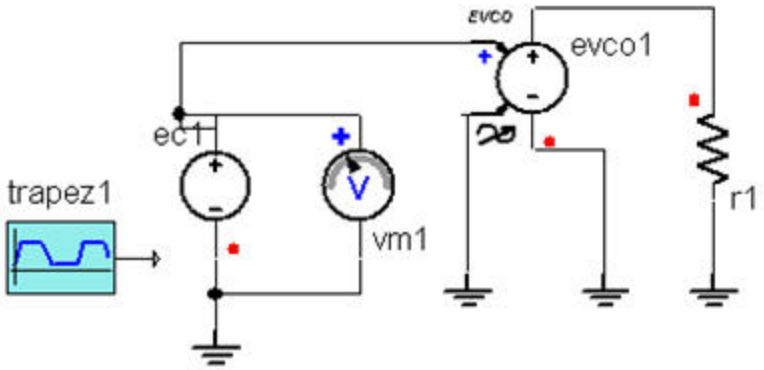


Figure 2. Application examples of the VHDL-AMS VCO Voltage Source model

Table 3. System Parameters

Component	Parameter	Value [unit]
Controlled Current Source ec1	fact	1
	quant	trapez1.val
VCO Voltage Source evco1	ampl	326 [V]
Trapezoidal Wave Source trapez1	ampl	311 [V]
	pwidth	10m [S]
	off	50 [S]
	trise	10m [S]
	tfall	10m [S]
Resistor r1	r	10 [Ohm]

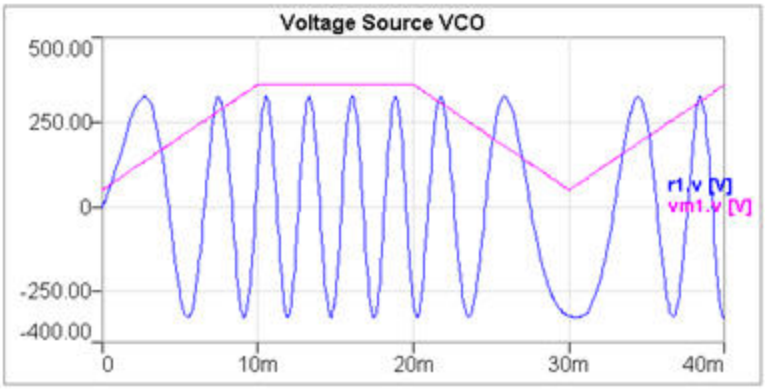


Figure 3. Simulation results-control current at voltmeter (vm1) and output voltage of VCO.

Top

References

Current Source

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents an independent current source. To define the source current, enter a numerical value, a variable, or expression in the parameter list.

The voltage of an ideal current source is arbitrary. Real current sources have always an internal resistance. If the internal resistance is essential for the simulation target, a corresponding resistance must be connected in parallel.

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

COUPL i ?InstanceName(@InstanceName):(@(Refbase)@(ID)) p := %0 , m := %1 (i := @i , ac_mag := @ac_mag , ac_phase := @ac_phase) DST: SIM(Type:=SimVHDL) SRC: DB

(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
p	Plus Terminal (with red point)	electrical
m	Minus Terminal	electrical

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
i	Current	real	1.0 [A]
ac_mag	Magnitude of EMF (AC)	real	1.0e-3 [A]
ac_ phase	Phase Shift of EMF (AC)	real	0.0 [deg]

Top

Example

This example shows a current source driving a linear resistor. In the example, two different ways of specifying the current value are demonstrated, a simple constant current case as shown in Figure 2, and an expression controlled case as shown in Figure 4.

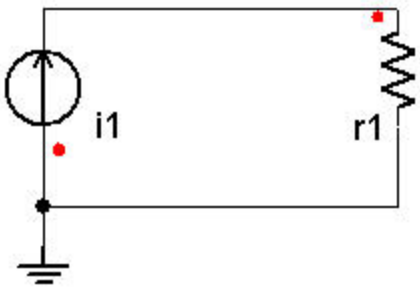


Figure 2. Application examples of the VHDL-AMS Current Source model

Table 3. System Parameters

Component	Parameter	Value [unit]

Current Source i1	i	10 [A]
Resistor r1	r	1k [Ohm]

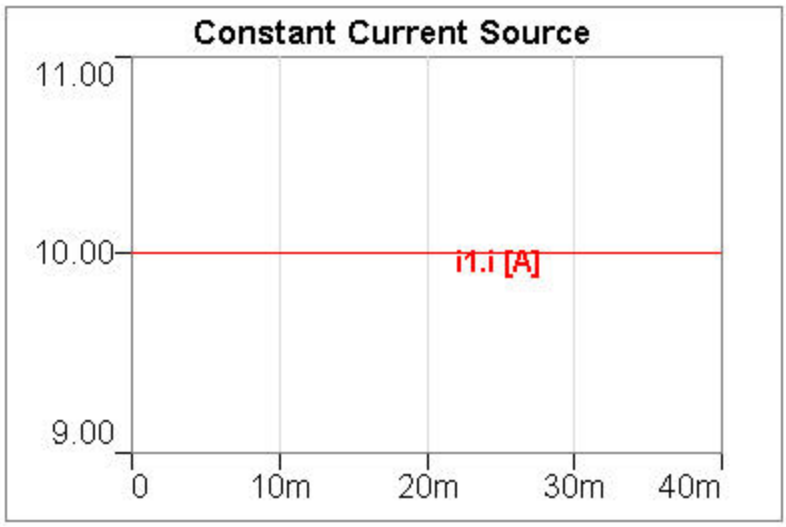


Figure 3. Simulation results-current output of current source i1.

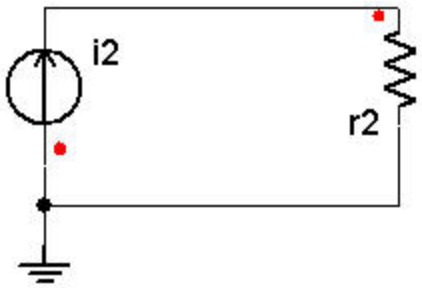


Figure 4. Application examples of the VHDL-AMS Current Source model with expression controlled output value

Table 4. System Parameters

Component	Parameter	Value [unit]
Current Source i2	i	2*t [V]
Resistor r2	r	1k [Ohm]

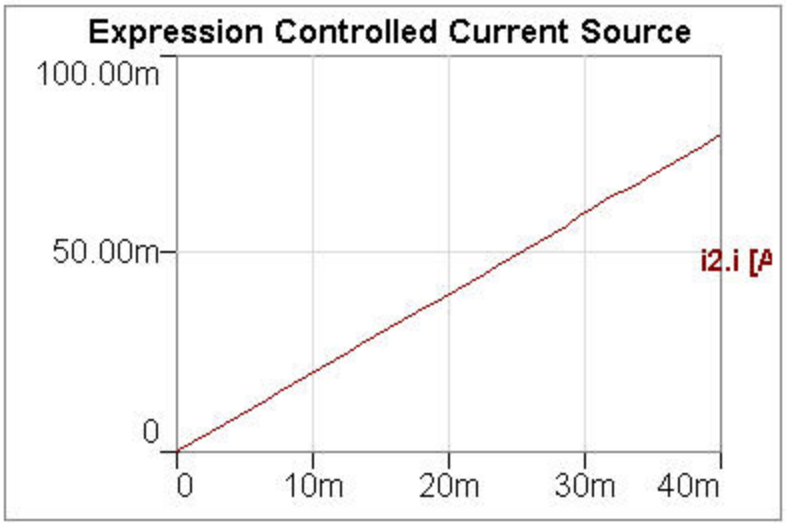


Figure 5. Simulation results-current output of current source i2 as a function of time.

Top

References

Controlled Current Source

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents a dependent current source. The value of the source is calculated from the controlling value multiplied by an arbitrary control coefficient.

To define controlling value and control coefficient, enter a numerical value, a variable, or expression in the parameter list.

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

COUPL ic ?InstanceName(@InstanceName):(@(Refbase)@(ID)) p := %0 , m := %1 (fact := @fact , quant := @quant) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
p	Plus Terminal (with red point)	electrical
m	Minus Terminal	electrical

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
quant	Control Quantity	real	0.0 [V or A]
fact	Factor	real	1.0 [/]
ac_mag	Magnitude of EMF (AC)	real	1.0e-3
ac_ phase	Phase Shift of EMF (AC)	real	0.0

Top

Example

This example demonstrates the use of both current and voltage control of a current source. In the example, an expression controlled current source (i3) is used to drive a resistor (r3). An ammeter and a voltage meter generate the control signals to drive the controlled sources (ic1 and ic2). The voltage generated by the controlled voltage sources are shown in Figure 3 and Figure 4.

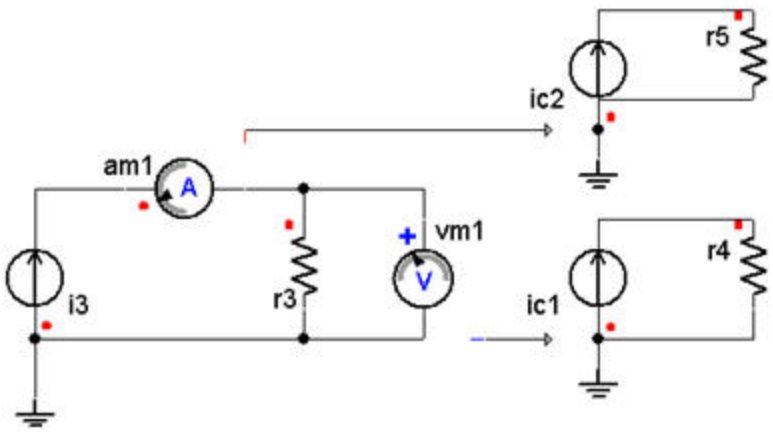


Figure 2. Application examples of the VHDL-AMS Controlled Voltage Source model

Table 3. System Parameters

Component	Parameter	Value [unit]
Current Source i3	i	2*t [A]
Controlled Current Source ic1	fact	3
	quant	vm1.v
Controlled Current Source ic2	fact	2
	quant	am1.i
Resistor r3	r	10 [Ohm]
Resistor r4/r5	r	1k [Ohm]

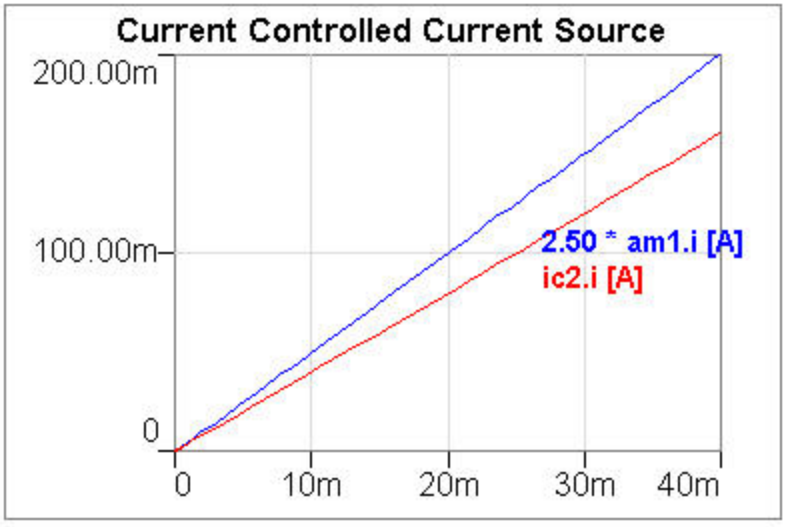


Figure 3. Simulation results-control current at ammeter (am1) and output of controlled current source (ic2).

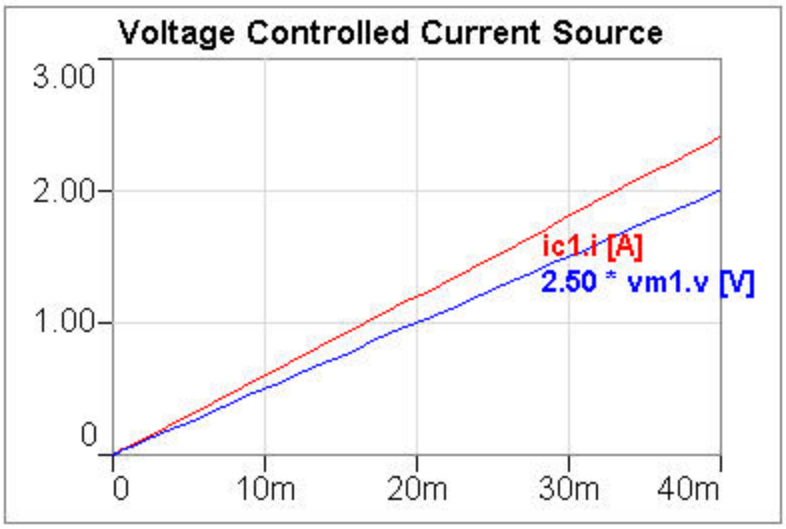


Figure 4. Simulation results-control voltage at voltmeter (vm1) and output of controlled current source (ic1).

Top

References

Voltage Controlled Oscillator Current Source

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

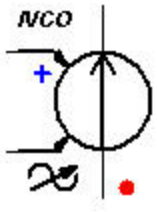


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents a voltage controlled Oscillator (current source). The VCO provides a sine wave with a frequency dependent on the input signal. The frequency is changed in relation 1:1, e.g. vcontrol=50V -> f=50Hz. Furthermore you can define amplitude and phase shift of the output sine wave and limit of the control signal.

The limit input voltage value, lmt interprets an input of 1 to indicate that the input voltage is limited and an input of 0 to indicate that the input voltage is not limited. If the lmt field is specified with a value of '1' the entry for the upper limit and lower limit is used for limiting the input voltage. This is represented as a static real value.

The upper limit (ul) and lower limit (ll) values are represented as static real values and used to limit the input voltage if lmt has a value of 1.0.

The amplitude value, ampl defines the amplitude of the output sine wave and is represented as a static real value.

The phase value, phase, specifies the phase shift of the output sine wave. The phase value needs to be specified in radians and is represented as a static real value.

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

COUPL ivco ?InstanceName(@InstanceName):(@(Refbase)@(ID)) p := %0 , m := %1 , fp := %2 , fm := %3 (phase := @phase , lmt := @lmt , ul := @ul , ll := @ll , ampl := @ampl , ac_mag := @ac_mag , ac_phase := @ac_phase) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
p	Plus Terminal	electrical
m	Minus Terminal	electrical
fp	Frequency Plus Terminal	electrical
fm	Frequency Minus Terminal	electrical

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
ul	Input Voltage Upper Limit	real	0.0 [V]
ll	Input Voltage Lower Limit	real	0.0 [V]
ampl	Amplitude	real	326.0 [V]
phase	Phase	real	0.0 [rad]
lmt	Limit Input Voltage Flag, 1=yes, 0=no	real	0.0
ac_mag	Magnitude of EMF (AC)	real	1.0E-3 [V]
ac_ phase	Phase Shift of EMF(AC)	real	0.0 [deg]

Top

Example

This example demonstrates the use of a VCO Current Source. A Voltage Controlled Oscillator (VCO) provides a sine wave output signal whose frequency depends upon the amplitude of the control signal, in this case a voltage input. In this example a controlled voltage source is driven by a trapezoidal source to provide the varying input to the VCO. The VCO output is a time varying current source.

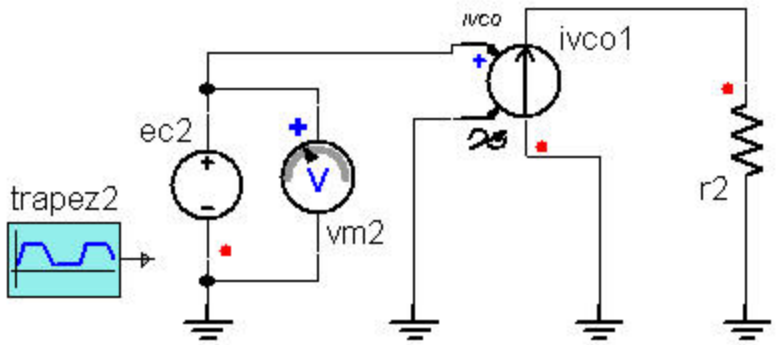


Figure 2. Application examples of the VHDL-AMS VCO Current Source model

Table 3. System Parameters

Component	Parameter	Value [unit]
Controlled Voltage Source ec2	fact	1
	quant	trapez2.val
VCO Voltage Source ivco1	ampl	-10 [V]
Trapezoidal Wave Source trapez2	ampl	100 [V]
	pwidth	10m [S]
	off	0 [S]
	trise	10m [S]
	tfall	10m [S]
Resistor r2	r	1k [Ohm]

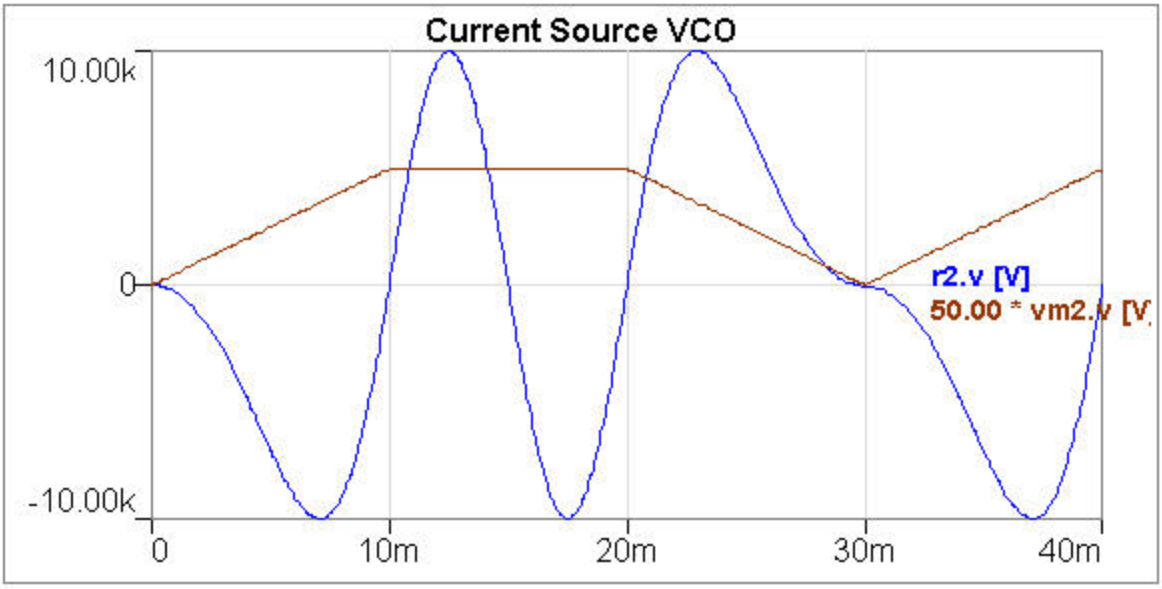


Figure 3. Simulation results-control voltage at voltmeter (vm1) and voltage across resistor (r2) resulting from current output of VCO Current Source.

Top

References

Switches

- Controlled Current Switch in VHDL-AMS (ccs)
- Controlled Voltage Switch in VHDL-AMS (cvs)
- Ideal Switch in VHDL-AMS (s)
- Ideal Transfer Switch in VHDL-AMS (ts)

Current Controlled Switch

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

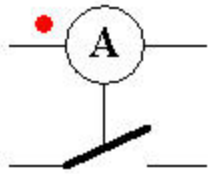


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represent an current controlled switch with on and off resistance.If the current is greater than threshold plus hysteresis, the controlled switch is on. If the current is lower than threshold minus hysteresis, the controlled switch is off. Because it is a non-ideal switch, you can define an On and Off resistance for the line connection.

To define the control value, enter a numerical value, a variable, or expression in the parameter list.

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

COUPL ccs ?InstanceName(@InstanceName):(@(Refbase)@(ID)) p := %0 , m := %1 , cp := %2 , cm := %3 (ron := @ron , roff := @roff , ithres := @ithres , ihyst := @ihyst) DST: SIM(Type:-:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
p	Plus Terminal (with red point)	electrical
m	Minus Terminal	electrical
cp	Control Plus Terminal (red point)	electrical
cm	Control Minus Terminal	electrical

Top

Parameters

Table 2

Name	Description	Data Type	Default Value[Unit]
ron	On State Resistance	real	1.0e-3 [Ohm]
roff	Off State Resistance	real	1.0e9 [Ohm]
ithres	Threshold Current	real	0.0 [A]
ihyst	Hysteresis Current	real	0.0 [A]

Top

Example

This example demonstrates the use of a Current Controlled Switch Model to switch a pulse source to a load resistor. The switch is controlled by a triangle wave source

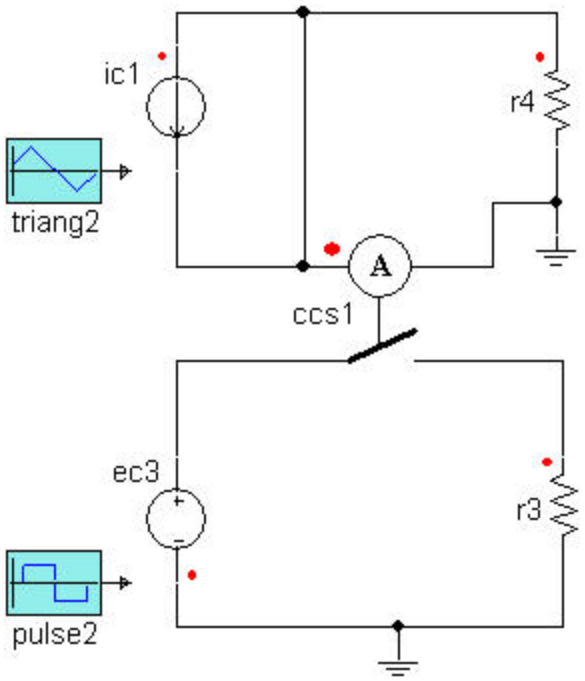


Figure 2. Application examples of the VHDL-AMS Current Controlled Switch model

Table 3. System Parameters

Component	Parameter	Value [unit]
Controlled Voltage Source ec3	fact	1
	quant	pulse2.val
Controlled Current Source ic1	fact	1
	quant	triang2.val
Current Controlled Switch ccs1	ithres	0.2 [A]
	ihyst	0.4 [A]
Pulse Wave Source pulse2	tdelay	0 [S]
	ampl	2 [V]
	freq	100 [Hz]

Triangular Wave Source triang2	tdelay	0 [S]
	ampl	1 [V]
	off	0 [S]
	freq	50 [Hz]
Resistor r3/r4	r	1k [Ohm]

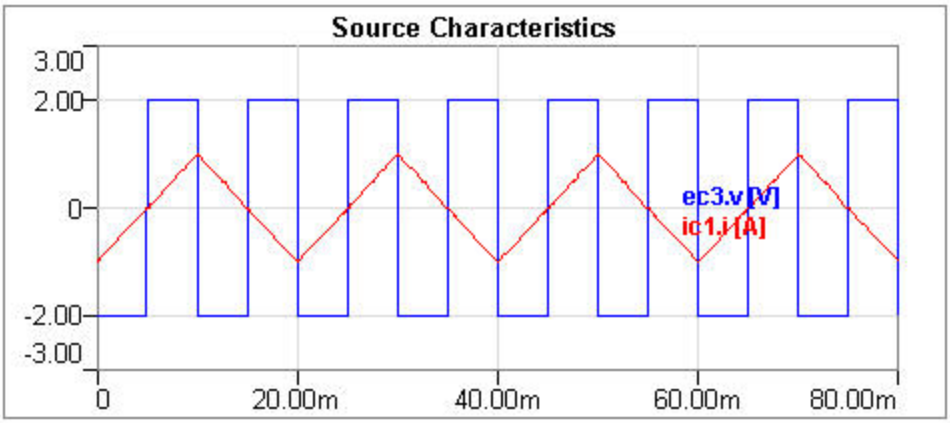


Figure 3. Simulation results-output from controlled sources (ec3/ic1).

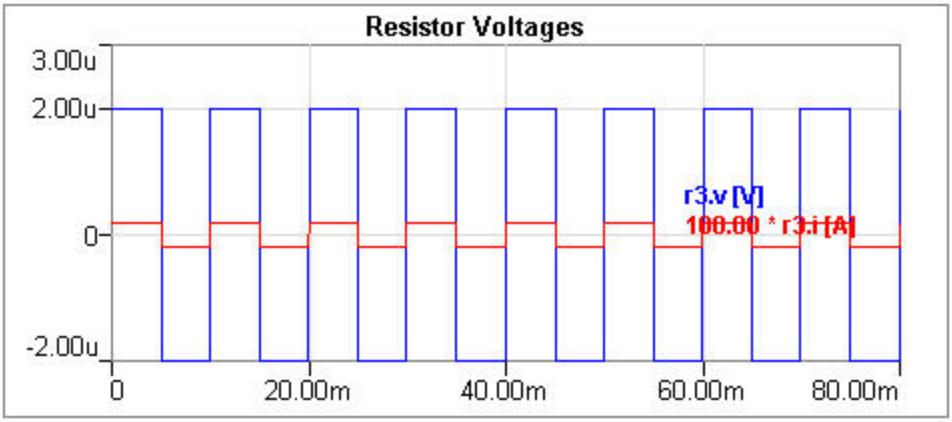


Figure 4. Simulation results-voltage across and current through resistor (r3) due to switching.

Top

References

Voltage Controlled Switch

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

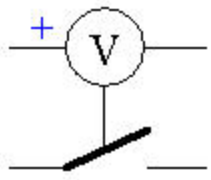


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represent an voltage controlled switch with On and Off resistance. If the voltage is greater than threshold plus hysteresis, the controlled switch is on. If the voltage is lower than threshold minus hysteresis, the controlled switch is off. Because it is a non-ideal switch, you can define an On and Off resistance for the line connection.

To define the control value, enter a numerical value, a variable, or expression in the parameter list.

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

COUPL cvs ?InstanceName(@InstanceName):(@(Refbase)@(ID)) p := %0 , m := %1 , cp := %2 , cm := %3 (ron := @ron , roff := @roff , vthres := @vthres , vhyst := @vhyst) DST: SIM(Type:-:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
p	Plus Terminal (with red point)	electrical
m	Minus Terminal	electrical
cp	Control Plus Terminal (with blue cross)	electrical
cm	Control Minus Terminal	electrical

Top

Parameters

Table 2

Name	Description	Data Type	Default Value[Unit]
ron	On State Resistance	real	1.0E-3 [Ohm]
roff	Off State Resistance	real	1.0E9 [Ohm]
vthres	Threshold Voltage	real	0.0 [V]
vhyst	Hysteresis Voltage	real	0.0 [V]

Top

Example

This example demonstrates the use of a Voltage Controlled Switch Model to switch a pulse source to a load resistor. The switch is controlled by a triangle wave source

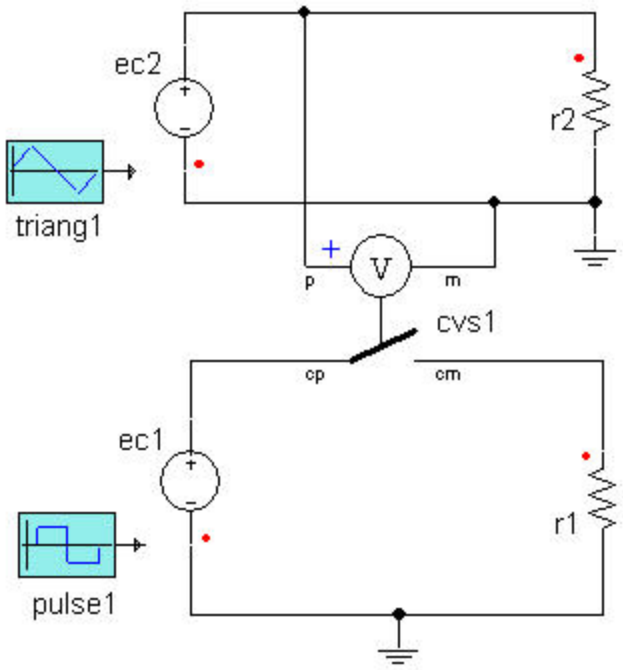


Figure 2. Application examples of the VHDL-AMS Voltage Controlled Switch model

Table 3. System Parameters

Component	Parameter	Value [unit]
Controlled Voltage Source ec1	fact	1
	quant	pulse1.val
Controlled Voltage Source ec2	fact	1
	quant	triang1.val
Voltage Controlled Switch cvs1	vthres	0.2 [V]
	vhyst	0.3 [V]
Pulse Wave Source pulse1	tdelay	0 [S]
	ampl	2 [V]
	freq	100 [Hz]

Triangular Wave Source triang1	tdelay	0 [S]
	ampl	1 [V]
	off	0 [S]
	freq	50 [Hz]
Resistor r1/r2	r	1k [Ohm]

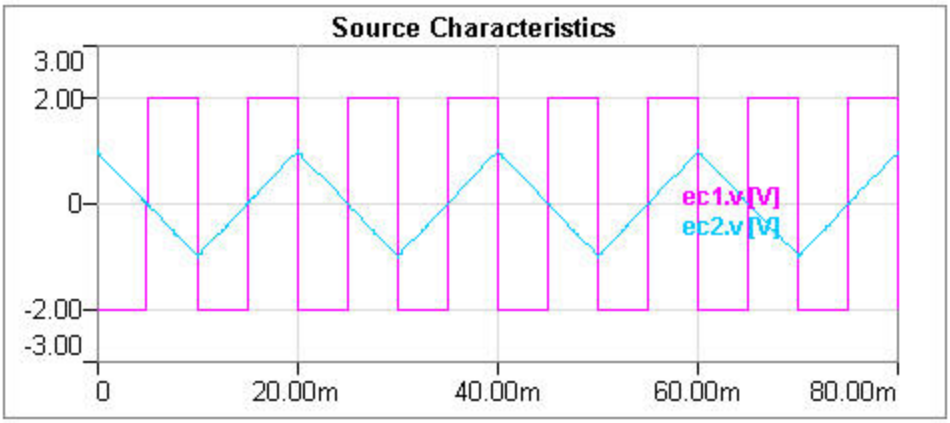


Figure 3. Simulation results-voltage output of controlled voltages sources (ec1/ec2).

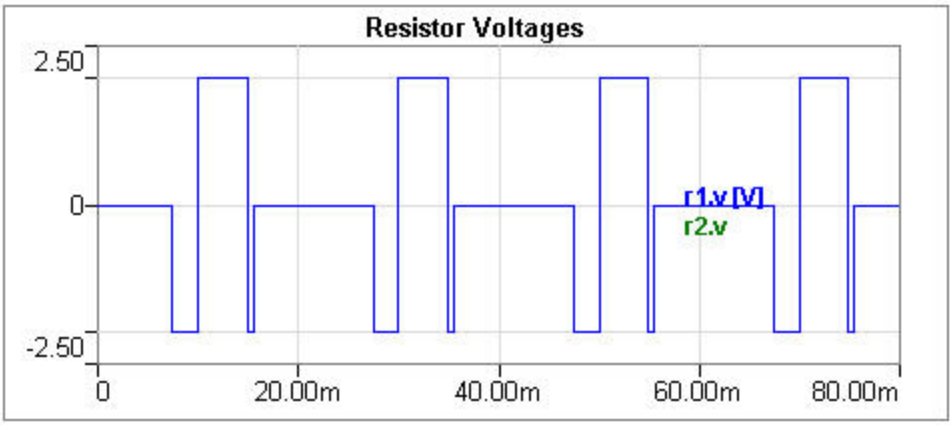


Figure 4. Simulation results-voltage across resistor (r1) due to switching.

Top

References

Ideal Switch

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents an ideal electrical switch. If the control signal is greater than '0', the two terminal nodes of the switch are connected. Because it is an ideal switch, the resistance of the line connection is zero. If the control signal is lower than or equal to '0', the line is disconnected and the resistance is infinite. A logical signal controls the state of the switch.

To define the control value, enter a numerical value, a variable, or expression in the parameter list.

Note: Since the switch is ideal, physically meaningful circuits must be obtained in both ON and OFF switching states.

The ideal switch is represented as an Animated Symbol; that means the switch symbol will change during the simulation depending on the corresponding control signal.

For more information see the following:

Enabling symbol animation in the main Twin Builder help.

Animate Dialog Box in the main Twin Builder help.

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

COUPL s ?InstanceName(@InstanceName):(@(Refbase)@(ID)) n1 := %0 , n2 := %1 (ctrl := @ctrl) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:-:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
n1	Node 1 (with red point)	electrical
n2	Node 2	electrical

Top

Parameters

Table 2

Name	Description	Data Type	Default Value[Unit]
ctrl	Control Signal	real	0.0

Top

Example

This example demonstrates the use of an Ideal Switch Model. A Pulse Wave source is used to control the Ideal Switch and alternately apply and remove voltage from a resistor/capacitor network.

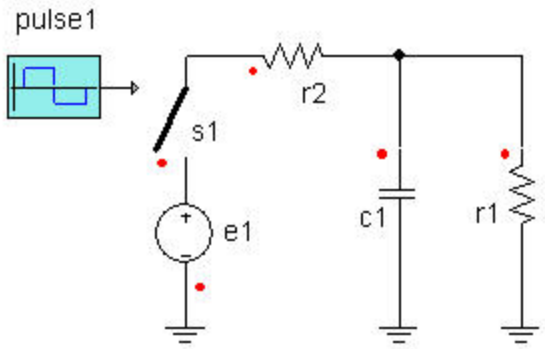


Figure 2. Application examples of the VHDL-AMS Ideal Switch model

Table 3. System Parameters

Component	Parameter	Value [unit]
Voltage Source e1	emf	1 [V]
Ideal Switch s1	ctrl	pulse1.val
Pulse Wave Source pulse1	tdelay	1m [S]
	ampl	311 [V]
	off	0 [S]
	freq	50 [Hz]
Resistor r1/r2	r	500 [Ohm]
Capacitor c1	c	1u [F]
	v0	0 [V]

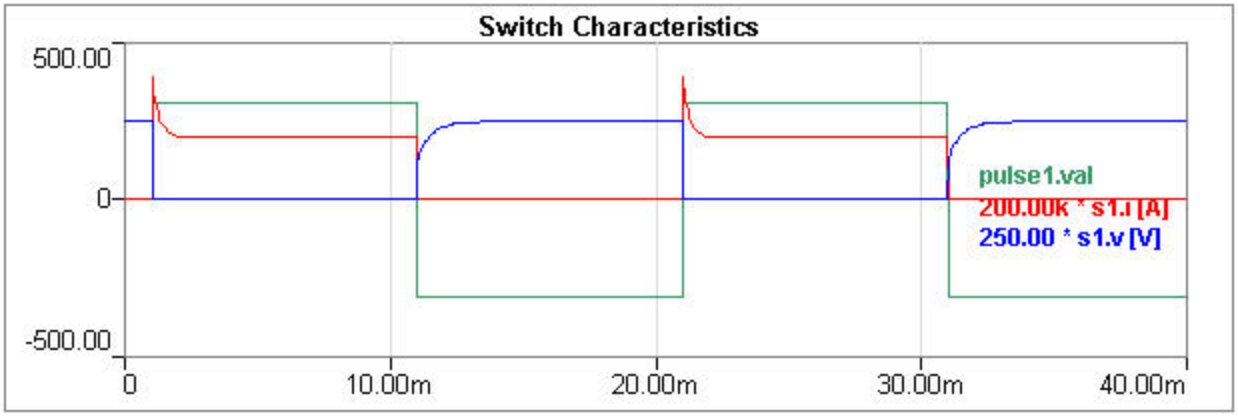


Figure 3. Simulation results-switch control voltage, and voltage and current at switch (s1).

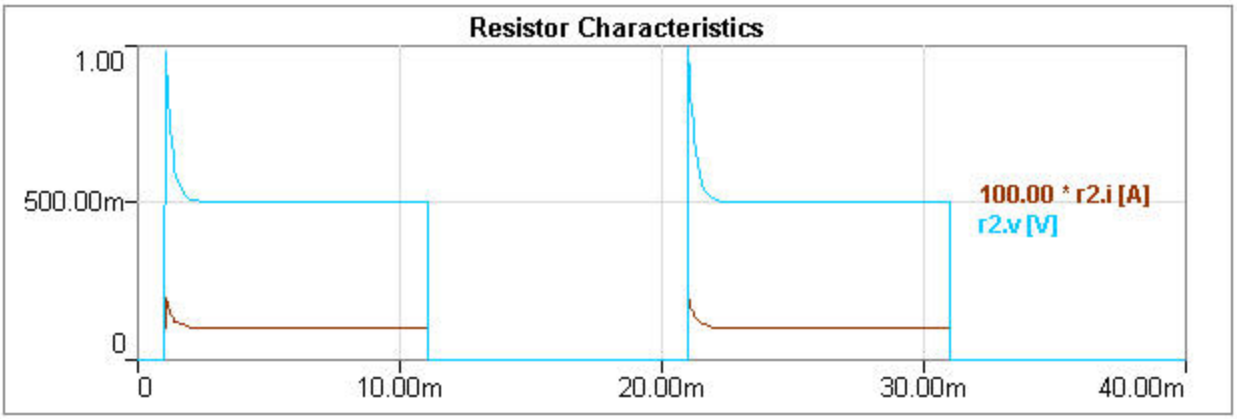


Figure 4. Simulation results-voltage and current at resistor (r2).

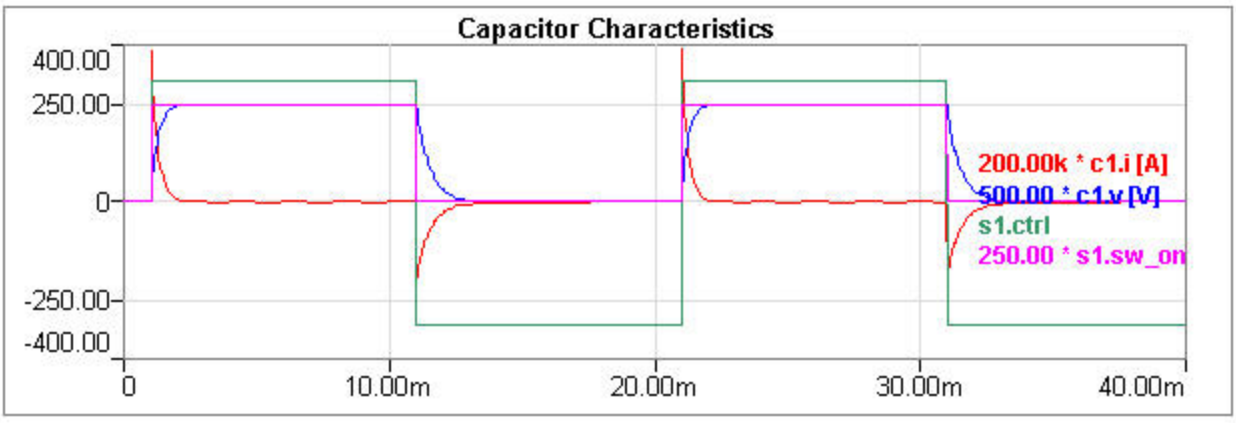


Figure 5. Simulation results-capacitor voltage and current, and switch control signal.

Top

References

Ideal Transfer Switch

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents an ideal electrical transfer switch. If the control signal is greater than '0', the nodes n1 and n2 are connected and the nodes n1 and n3 are disconnected. Because it is an ideal switch, the resistance of the line connection is zero and the resistance of the dis-connection is infinite. If the control signal is lower than or equal to '0', the line between n1 and n3 is connected.

To define the control value, enter a numerical value, a variable, or expression in the parameter list.

Note: Since the switch is ideal, physically meaningful circuits must be obtained in both ON and OFF switching states.

The ideal switch is represented as an Animated Symbol; that means the switch symbol will change during the simulation depending on the corresponding control signal.

For more information see the following:

Enabling symbol animation in the main Twin Builder help.

Animate Dialog Box in the main Twin Builder help.

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

COUPL ts ?InstanceName(@InstanceName):(@(Refbase)@(ID)) n1 := %0 , n2 := %1 , n3 := %2 (ctrl := @ctrl) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:-:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
n1	Node 1 (with red point)	electrical
n2	Node 2	electrical
n3	Node 3	electrical

Top

Parameters

Table 2

Name	Description	Data Type	Default Value[Unit]
ctrl	Control Signal	real	0.0

Top

Example

This example demonstrates the use of an Ideal Transfer Switch Model to switch between two sinusoidal sources. A Pulse Wave source is used to control the Ideal Transfer Switch and alternately apply sources sine1 and sine2 a resistor load.

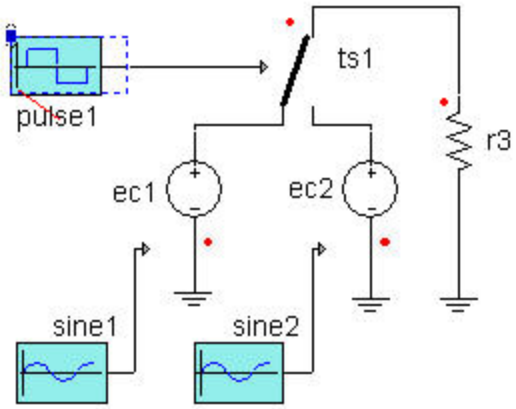


Figure 2. Application examples of the VHDL-AMS Ideal Transfer Switch model

Table 3. System Parameters

Component	Parameter	Value [unit]
Controlled Voltage Source ec1	fact	1
	quant	sine1.val
Controlled Voltage Source ec2	fact	1
	quant	sine2.val
Ideal Switch s1	ctrl	pulse1.val
Pulse Wave Source pulse1	tdelay	0 [S]
	ampl	1 [V]
	off	0 [S]
	freq	100 [Hz]
Sine Wave Source sine1	tdelay	0 [S]
	ampl	32 [V]
	off	0 [S]
	freq	500 [Hz]
Sine Wave Source sine2	tdelay	0 [S]
	ampl	32 [V]
	off	0 [S]
	freq	5000 [Hz]
Resistor r3	r	1k [Ohm]

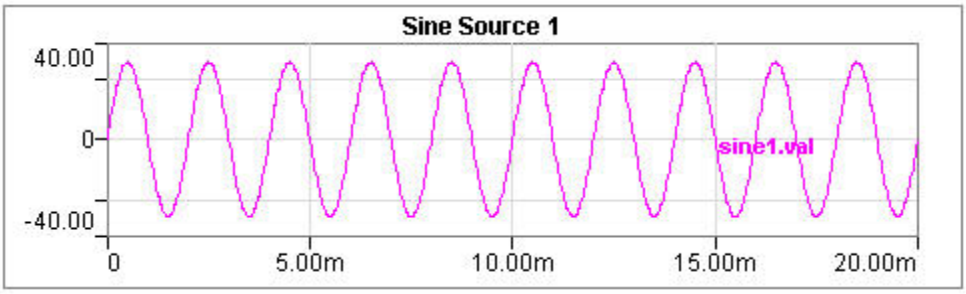


Figure 3. Simulation results-sine 1 voltage output (sine1/ec1).

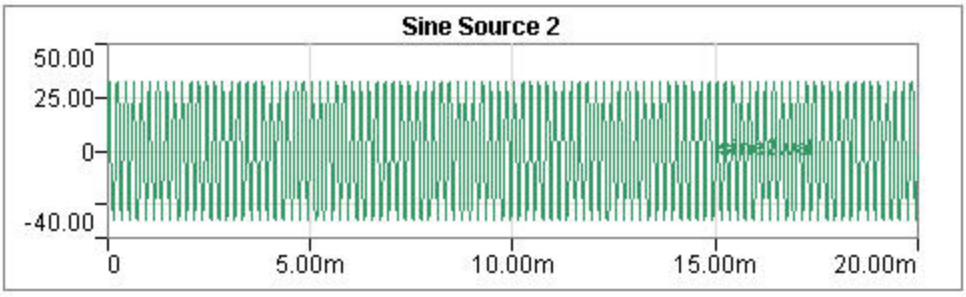


Figure 4. Simulation results-voltage output of sine2.

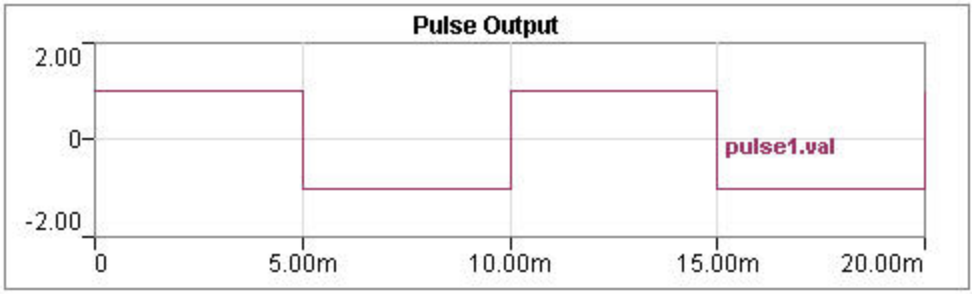


Figure 5. Simulation results-pulse control signal to Ideal Transfer Switch.

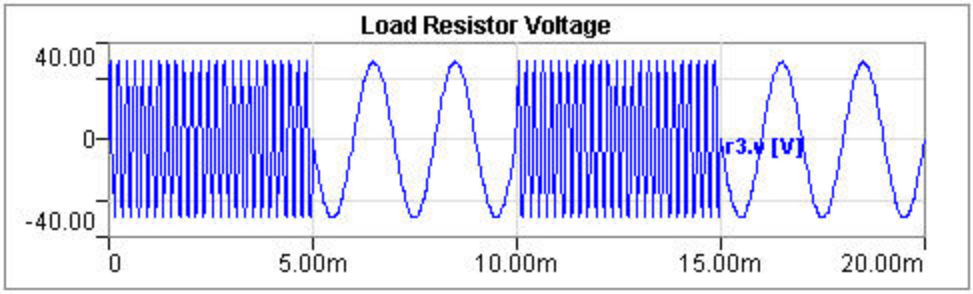


Figure 6. Simulation results-voltage across the load resistor.

Top

References

Transformers

• Single-Phase Systems

• Three-Phase Systems

Twin Builder Transformer Models

Transformer components, available in the *Circuit* folder of the **Basic Elements VHDLAMS library**, are characterized by any energy flow direction, a free number of galvanically separate windings, the exclusion of DC-current transmission and the consideration of losses.

In the Spiro model, every limb of the transformer is represented by one inductance. Each winding, that influences the flux through the limb, contributes to the current flowing through the inductance. Usually the ratio flux/current is different for the winding of the limb because of different numbers of turns or, in the case of three-phase transformers, different positions of the windings. The current of only one winding can flow directly through the inductance. The other windings contribute by coupling factors.

Single Phase

The circuit shown in Figure 1 shows the equivalent circuit for a single phase transformer. The current through the primary winding contributes directly to the current through the main inductance. Therefore the main inductance is calculated with respect to the primary winding.

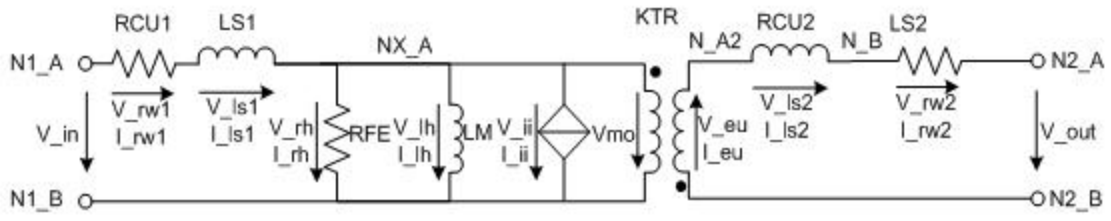


Figure 1. Equivalent circuit of the Single Phase Transformer.

Because of primary and secondary side stray inductances, coupling factors are introduced. The coupling factor for the secondary side is determined by the ratio of the number of turns only. Therefore the current through the secondary winding controls the current source at the primary side with the control factor KTR which is the number-of-turns ratio.

The same relationships apply for the voltage induced at primary and secondary side respectively. Since the main inductance is calculated for the primary winding and its current is calculated with respect to the primary side, the induced voltage at the primary side equals the voltage across the main inductance. The induced voltage at the secondary side equals the voltage across the main inductance multiplied by the number-of-turns ratio KTR.

Model Extension for Three-Phase Systems

To model a three-phase system, the coupling between all six windings has to be considered. The model for the single-phase transformer has to be used three times. For every phase the

influence of two more primary windings and two more secondary windings has to be modeled. This influence is modeled as for the single-phase transformer using controlled current and voltage sources.

Assuming the same number of turns for all primary windings the coupling between the primary windings depends on the geometry only. The coupling of a secondary winding to the primary winding of the same phase depends on the number-of-turns ratio again (as for single-phase transformers).The coupling from a secondary side to a primary side of another phase depends on geometry and number-of-turns ratio.

Three-Phase Transformer Twin Builder Models

Twin Builder provides a separate model for primary and secondary windings, which can be adjusted to your needs individually. The coupling factors can be determined by a FEM-analysis or by an estimation of the magnetic resistances based on the core geometry.

There are also two parameterized models for three-phase transformers of small and large power. The parameters were determined by analyzing existing transformers with so-called EI-core.

See also Six-Winding Transformer(large power)

See also Six-Winding Transformer(small power)

Single-Phase Systems

- Primary Side of a Two-Winding Transformer (tfr1p1)
- Secondary Side of a Two-Winding Transformer (tfr1p2)
- Linear Two-Winding Transformer (tfr1p2w)

Primary Side of a Two-Winding Transformer

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

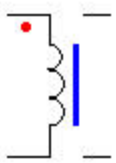


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Input/Output Quantities

• Example

• References

Description

The component represents the primary side of a transformer based on the equivalent circuit shown in Figure 2. For each coil, make sure a network path exists to the ground node. Within the dialog box, you can define the main and leakage inductances, the equivalent resistances for iron losses, the winding resistances for each side, and the initial values of the current. To define the parameter values, enter a numerical value, a variable, or expression in the corresponding field of parameter list.

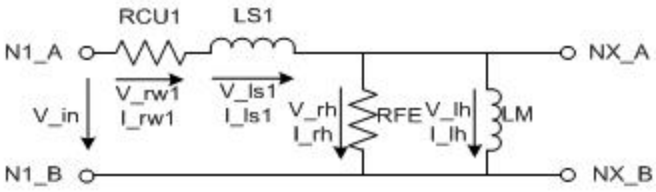


Figure 2. Equivalent Circuit of Primary Side of a Linear Transformer

Note: The terms primary and secondary are not intended in the strict sense. It means only to combine exactly one macro of the primary side with at least one macro of the secondary side.

Top

Assumptions and Limitations

This model does not take into account the magnetizing characteristics of the core.

Top

Mathematical Description

Equations that are used to describe the linear two-winding transformer primary side are listed as follows:

$$V\_rw1 = I\_rw1 \cdot R_{cu1}$$

$$P_{si}\_ls1 = I\_ls1 \cdot L_{ls1}$$

$$V\_ls1 = \frac{d\,P_{si}\_ls1}{d\,t}$$

$$P_{si}\_lh = I\_lh \cdot L_M$$

$$V\_lh = \frac{d\,P_{si}\_lh}{d\,t}$$

Top

Netlist Syntax

COUPL tfr1p1 ?InstanceName(@InstanceName):(@(Refbase)@(ID)) n1_a := %0 , n1_b := %1 , nx_a := %2 , nx_b := %3 (ls1 := @ls1 , rfe := @rfe , i0_ls1 := @i0_ls1 , lm := @lm , rcu1 := @rcu1) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:-:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type

N1_A	Positive Node on the Primary Side	electrical
N1_B	Negative Node on the Primary Side	electrical
NX_A	Positive Node at the Connection Point	electrical
NX_B	Negative Node at the Connection Point	electrical

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
lm	Magnetizing Inductance	real	0.1 [H]
ls1	Leakage Inductance Primary Side	real	0.001 [H]
rfe	Equivalent Resistance for Iron Losses	real	1.0e18 [Ohm]
rcu1	Winding Resistance Primary Side	real	1.0e-6 [Ohm]
I0_LS1	Initial Current Primary Side	real	0 [A]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
V1	Voltage Primary Side [V]		real

Top

Example

See Single Phase Transformer Example

Top

References

Secondary Side of a Two-Winding Transformer

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

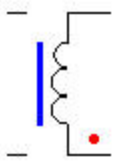


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents the secondary side of a transformer based on the equivalent circuit shown in Figure 2. For each coil make sure a network path exists to the ground node. Within the dialog users can define the main and leakage inductances, the equivalent resistances for iron losses, the winding resistances for each side, and the initial values of the current. To define the parameter values, enter a numerical value, a variable, or expression in the corresponding field of parameter list.

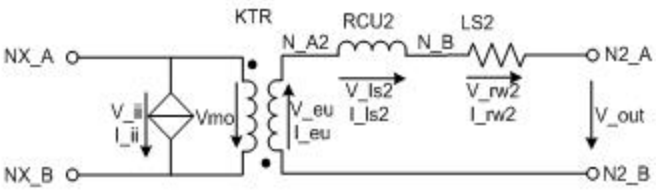


Figure 2. Equivalent circuit of the Secondary Side of a Linear Transformer

Note: The terms primary and secondary are not intended in the strict sense. It simply means to combine exactly one macro of the primary side with at least one macro of the secondary side.

Top

Assumptions and Limitations

This model does not take into account the magnetizing characteristics of the core.

Top

Mathematical Description

Equations that are used to describe the linear two-winding transformer secondary side are listed as follows:

$$I\_{ii} = K_{TR} \cdot I\_{eu} = K_{TR} \cdot I\_{ls2}$$

$$V\_{eu} = v_{m0} \cdot K_{TR}$$

$$P_{si}\_{ls2} = I\_{ls2} \cdot L_{ls2}$$

$$V\_{ls2} = \frac{d\, P_{si}\_{ls2}}{d\, t}$$

$$V\_{rw2} = I\_{rw2} \cdot R_{cu2}$$

Top

Netlist Syntax

COUPL tfr1p2 ?InstanceName(@InstanceName):(@(Refbase)@(ID)) nx_a := %0 , nx_b := %1 , n2_a := %2 , n2_b := %3 (ls2 := @ls2 , ktr := @ktr , i0_ls2 := @i0_ls2 , rcu2 := @rcu2) DST: SIM (Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:-:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
NX_A	Positive Node at the Connection Point	electrical
NX_B	Negative Node at the Connection Point	electrical
N2_A	Positive Node Secondary Side	electrical
N2_B	Negative Node Secondary Side	electrical

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
LS2	Leakage Inductance Secondary Side	real	0.001 [H]
RCU2	Winding Resistance Secondary Side	real	1.0e-6 [Ohm]
KTR	Winding Ratio	real	1.0 [/]
I0_IS2	Initial Current Secondary Side	real	0 [A]

Top

Example

See Single Phase Transformer Example

Top

References

Linear Two-Winding Transformer

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

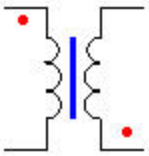


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents a transformer based on the equivalent circuit shown in Figure 2. For each coil make sure a network path exists to the ground node. Within the dialog users can define the main and leakage inductances, the equivalent resistances for iron losses, the winding resistances for each side, and the initial values of the current. To define the parameter values, enter a numerical value, a variable, or expression in the corresponding field of parameter list. This model does not take into account the magnetizing characteristics of the core.

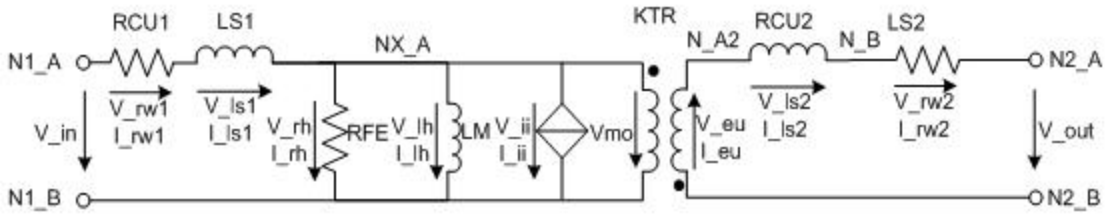


Figure 2. Equivalent circuit of the Linear Two-Winding Transformer

Top

Assumptions and Limitations

This model does not take into account the magnetizing characteristics of the core.

Top

Mathematical Description

Equations that are used to describe the linear two-winding transformer are listed as follows:

$$V_{\_rw1} = I_{\_rw1} \cdot R_{cu1}$$

$$P_{si\_ls1} = I_{\_ls1} \cdot L_{ls1}$$

$$V_{\_ls1} = \frac{d\ P_{si\_ls1}}{d\ t}$$

$$P_{si\_lh} = I_{\_lh} \cdot L_M$$

$$V_{\_lh} = \frac{d\ P_{si\_lh}}{d\ t}$$

$$V_{\_rh} = I_{\_rh} \cdot R_{FE}$$

$$I_{\_ii} = K_{TR} \cdot I_{\_eu} = K_{TR} \cdot I_{\_ls2}$$

$$V_{\_eu} = v_{m0} \cdot K_{TR}$$

$$P_{si\_ls2} = I_{\_ls2} \cdot L_{ls2}$$

$$V_{\_ls2} = \frac{d\ P_{si\_ls2}}{d\ t}$$

$$V_{\_rw2} = I_{\_rw2} \cdot R_{cu2}$$

Top

Netlist Syntax

COUPL tfr1p2w ?InstanceName(@InstanceName):(@(Refbase)@(ID)) n1_a := %0 , n1_b := %1 , n2_a := %2 , n2_b := %3 (ls1 := @ls1 , ls2 := @ls2 , rfe := @rfe , ktr := @ktr , i0_ls1 := @i0_ ls1 , i0_ls2 := @i0_ls2 , lm := @lm , rcu1 := @rcu1 , rcu2 := @rcu2) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
N1_A	Positive Node on the Primary Side	electrical
N1_B	Negative Node on the Primary Side	electrical
N2_A	Positive Node on the Secondary Side	electrical
N2_B	Negative Node on the Primary Side	electrical

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
LM	Magnetizing Inductance	real	0.1 [H]
LS1	Leakage Inductance Primary Side	real	0.001 [H]
LS2	Leakage Inductance Secondary Side	real	0.001 [H]
RFE	Equivalent Resistance for Iron Losses	real	1.0e18 [Ohm]
RCU1	Winding Resistance Primary Side	real	1.0e-6 [Ohm]
RCU2	Winding Resistance Secondary Side	real	1.0e-6 [Ohm]
KTR	Winding Ratio	real	1.0 [/]
I0_LS1	Initial Current Primary Side	real	0 [A]
I0_LS2	Initial Current Secondary Side	real	0 [A]

Top

Example

This example demonstrates the application of a simple Two-Winding transformer to deliver a signal to a load. The example is set up with a winding ratio of 1:1.

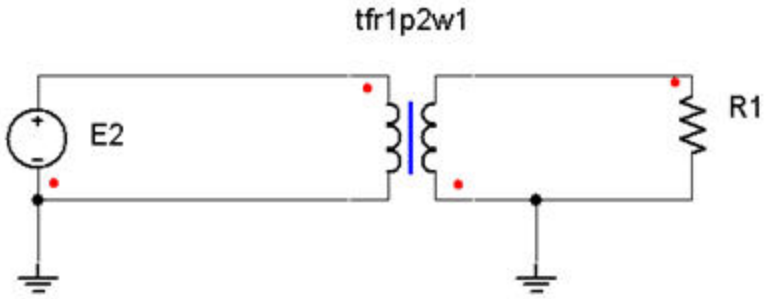


Figure 3. Application examples of the VHDL-AMS Linear Two-Winding Transformer model

Table 4. System Parameters

Component	Parameter	Value [unit]
Time Controlled Voltage Source (Sinusoidal) E2	AMPL	0.326k [V]
	Linear Two-Winding Transformer tfr1p2w1	ktr .- 0
	Resistor R1	

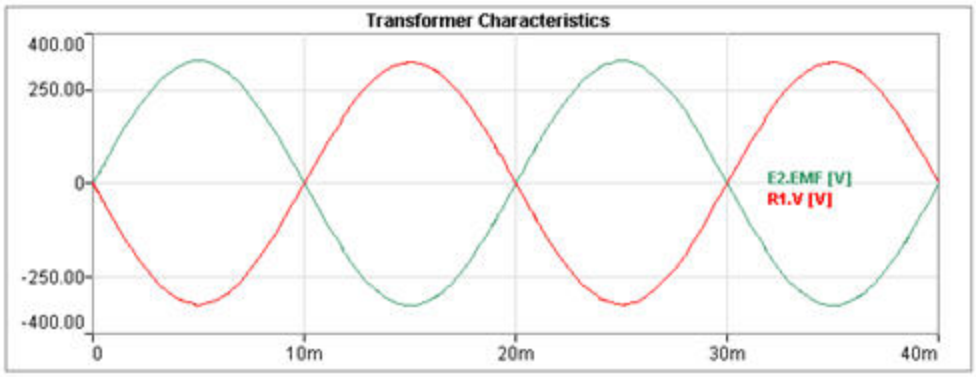


Figure 4. Simulation results-input to transformer (E2.EMF) and output voltage to resistor (R1.V).

Top

References

Single-Phase Transformer Example

This example demonstrates the use of the Primary Side and Secondary Side Transformer models to implement a transformer with a 2:1 winding ratio.

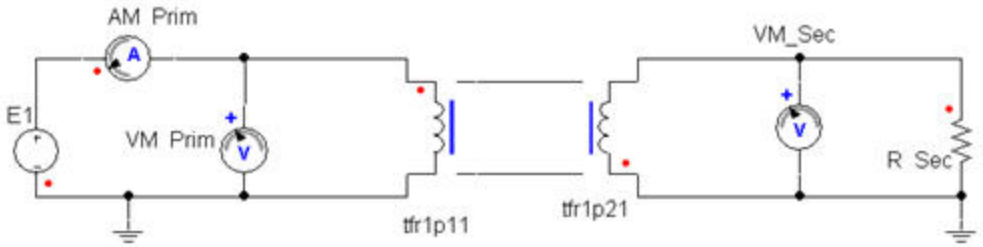


Figure 1. Application examples of the VHDL-AMS Transformer Primary-Side and Secondary-Side models

Table 1. System Parameters

Component	Parameter	Value [unit]
Time Controlled Voltage Source (Sinusoidal) E1	AMPL	0.326k [V]
	Transformer Primary Side tfr1p11	ls=1.0e-6 H
		ls=2.0e-6 H Transformer Secondary Side tfr1p21

Resistor R_Sec		r [- O- h- m-]

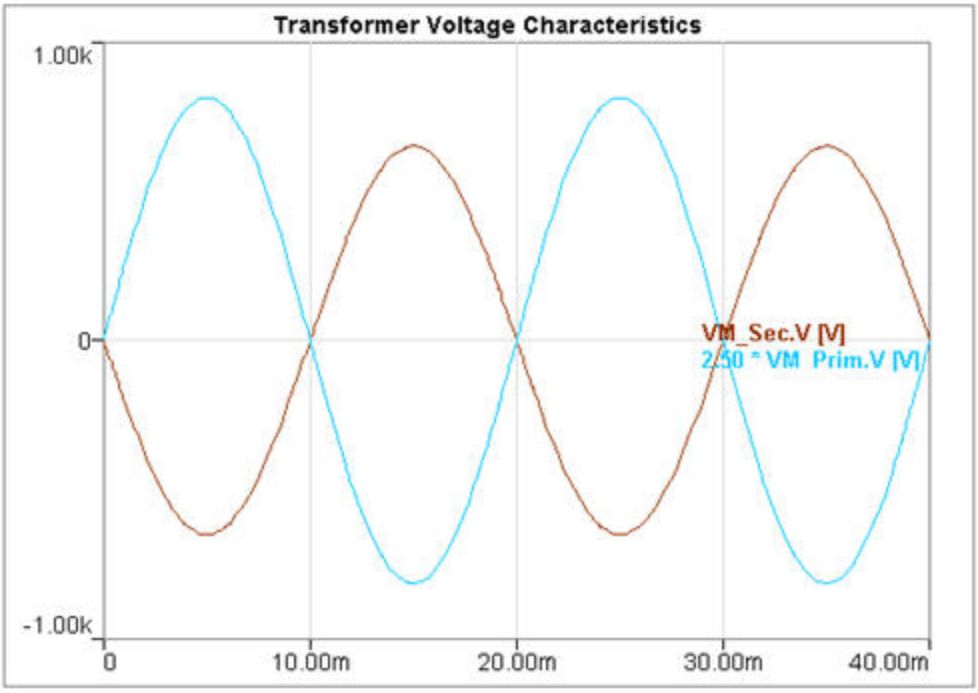


Figure 2. Simulation results-voltage on primary and secondary sides of transformer.

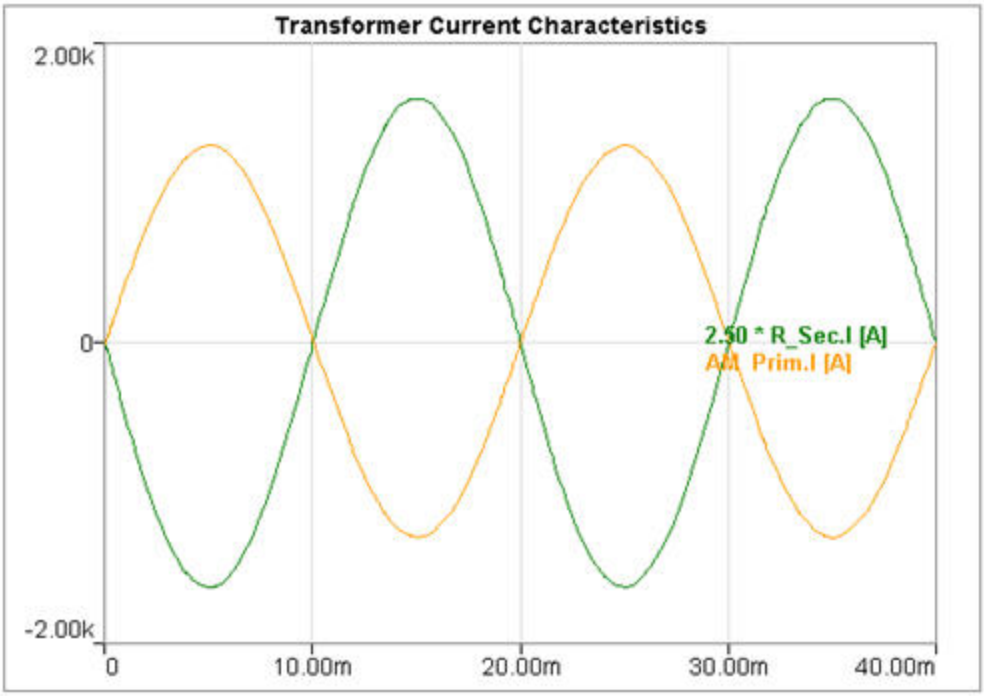


Figure 3. Simulation results-current on primary and secondary sides of transformer.

Three-Phase Systems

- Primary Side of a Six-Winding Transformer (tfr3p1)
- Secondary Side of a Six-Winding Transformer (tfr3p2)
- Linear Three-Phase Six-Winding Transformer - Large Power (tfr3p6wl)
- Linear Three-Phase Six-Winding Transformer - Small Power (tfr3p6ws)

Primary Side of a Six-Winding Transformer

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents the primary side of linear six-winding transformers based on the equivalent circuit shown in Figure 2. The use of this macro makes only sense with at least one macro of the secondary side. Using the Primary Side and Secondary Side of a Six-winding-Transformer allows a simple construction of a Three-Phase Transformer with more windings.

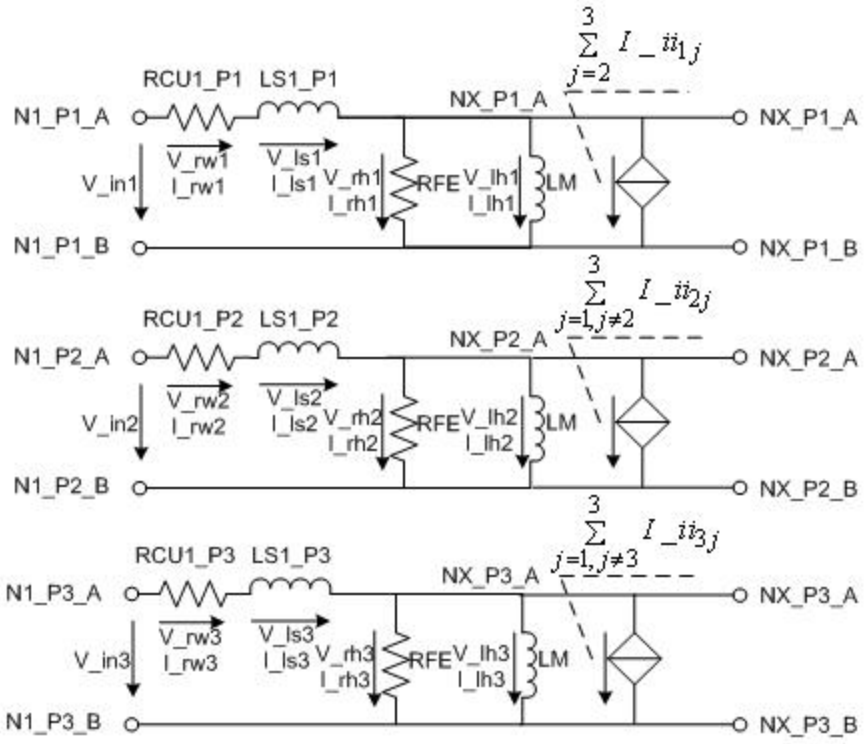


Figure 2. Equivalent circuit of the linear six-winding transformer primary side

Note: The terms primary and secondary are not intended in the strict sense. It simply means to combine exactly one macro of the primary side with at least one macro of the secondary side.

For each coil make sure a network path exists to the ground node. Within the dialog users can define the main and leakage inductances, the equivalent resistances for iron losses, the winding resistances for each side, and the initial values of the current. To define the parameter values, enter a numerical value, a variable, or expression in the corresponding field of parameter list.

Top

Assumptions and Limitations

This model does not take into account the magnetizing characteristics of the core.

Top

Mathematical Description

Equations that are used to describe the linear six-winding transformer are listed as follows:

Table 1

Phase 1	Phase 2	Phase 3
$V\_rw1 = I\_rw1 \cdot R_{cu1}\_p1$	$V\_rw2 = I\_rw2 \cdot R_{cu1}\_p2$	$V\_rw1 = I\_rw1 \cdot R_{cu1}\_p3$
$P_{si}\_ls1 = I\_ls1 \cdot L_{ls1}\_p1$	$P_{si}\_ls2 = I\_ls2 \cdot L_{ls1}\_p2$	$P_{si}\_ls5 = I\_ls5 \cdot L_{ls1}\_p3$
$V\_ls1 = \frac{d\ P_{si}\_ls1}{d\ t}$	$V\_ls2 = \frac{d\ P_{si}\_ls2}{d\ t}$	$V\_ls5 = \frac{d\ P_{si}\_ls5}{d\ t}$
$P_{si}\_lh1 = I\_lh1 \cdot L_M$	$P_{si}\_lh2 = I\_lh2 \cdot L_M$	$P_{si}\_lh3 = I\_lh3 \cdot L_M$
$V\_lh1 = \frac{d\ P_{si}\_lh1}{d\ t}$	$V\_lh2 = \frac{d\ P_{si}\_lh2}{d\ t}$	$V\_lh3 = \frac{d\ P_{si}\_lh3}{d\ t}$
$V\_rh1 = I\_rh1 \cdot R_{FE}$	$V\_rh2 = I\_rh2 \cdot R_{FE}$	$V\_rh3 = I\_rh3 \cdot R_{FE}$
$I\_ii12 = K_{TR1} \cdot I\_rw2$	$I\_ii21 = K_{TR3} \cdot I\_rw1$	$I\_ii31 = K_{TR2} \cdot I\_rw1$
$I\_ii12 = K_{TR2} \cdot I\_rw3$	$I\_ii23 = K_{TR3} \cdot I\_rw3$	$I\_ii32 = K_{TR1} \cdot I\_rw2$

Where K_{TR1}=0.5, K_{TR2}=0.25, K_{TR3}=0.75, for small-power transformers, and K_{TR1}=0.5, K_{TR2}=0.333, K_{TR3}=0.667, for large-power transformers

Top

Netlist Syntax

COUPL tfr3p1 ?InstanceName(@InstanceName):(@(Refbase)@(ID)) n1_p1_a := %0 , n1_p1_b := %1 , n1_p2_a := %2 , n1_p2_b := %3 , n1_p3_a := %4 , n1_p3_b := %5 , nx_p1_a := %6 , nx_p1_b := %7 , nx_p2_a := %8 , nx_p2_b := %9 , nx_p3_a := %10 , nx_p3_b := %11 (ls1_p1 := @ls1_p1 , ls1_p2 := @ls1_p2 , ls1_p3 := @ls1_p3 , rfe := @rfe , ktr1 := @ktr1 , ktr2 := @ktr2 , ktr3 := @ktr3 , i0_ls1 := @i0_ls1 , i0_ls2 := @i0_ls2 , i0_ls3 := @i0_ls3 , lm := @lm , rcu1_p1 := @rcu1_p1 , rcu1_p2 := @rcu1_p2 , rcu1_p3 := @rcu1_p3) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 2

Name	Port/Terminal Description	Nature/Data Type
N1_P1_A	Positive node on the primary side of phase 1	electrical
N1_P1_B	Negative node on the primary side of phase 1	electrical
N1_P2_A	Positive node on the primary side of phase 2	electrical
N1_P2_B	Negative node on the primary side of phase 2	electrical
N1_P3_A	Positive node on the primary side of phase 3	electrical

N1_P3_B	Negative node on the primary side of phase 3	electrical
NX_P1_A	Positive node at the connection point of phase 1	electrical
NX_P1_B	Negative node at the connection point of phase 1	electrical
NX_P2_A	Positive node at the connection point of phase 2	electrical
NX_P2_B	Negative node at the connection point of phase 2	electrical
NX_P3_A	Positive node at the connection point of phase 3	electrical
NX_P3_B	Negative node at the connection point of phase 3	electrical

Top

Parameters

Table 3

Name	Description	Data Type	Default Value[Unit]
LS1_P1	Leakage Inductance Primary Side of Phase 1	real	0.001 [H]
LS1_P2	Leakage Inductance Primary Side of Phase 2	real	0.001 [H]
LS1_P3	Leakage Inductance Primary Side of Phase 3	real	0.001 [H]
RCU1_P1	Winding Resistance Primary Side of Phase 1	real	1.0e-6 [Ohm]
RCU1_P2	Winding Resistance Primary Side of Phase 2	real	1.0e-6 [Ohm]
RCU1_P3	Winding Resistance Primary Side of Phase 3	real	1.0e-6 [Ohm]
LM	Magnetizing Inductance	real	0.1 [H]
RFE	Equivalent Resistance for Iron Losses	real	1.0e18 [Ohm]
KTR1	Coupling factor middle to outer limb	real	0.5
KTR2	Coupling factor outer to outer limb	real	0.25
KTR3	Coupling factor outer to middle limb	real	0.75
I0_IS1	Initial Current Primary Side of Phase 1	real	0 [A]
I0_IS2	Initial Current Primary Side of Phase 2	real	0 [A]
I0_IS3	Initial Current Primary Side of Phase 3	real	0 [A]

Top

Example

See Three-Phase Six-Winding Transformer Example

Top

References

Secondary Side of a Six-Winding Transformer

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

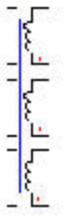


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents the secondary side of linear six-winding transformers based on the equivalent circuit shown in Figure 2. The use of this macro makes only sense with at least one macro of the secondary side. Using the Primary Side and Secondary Side of a Six-winding-Transformer allows a simple construction of a Three-Phase Transformer with more windings.

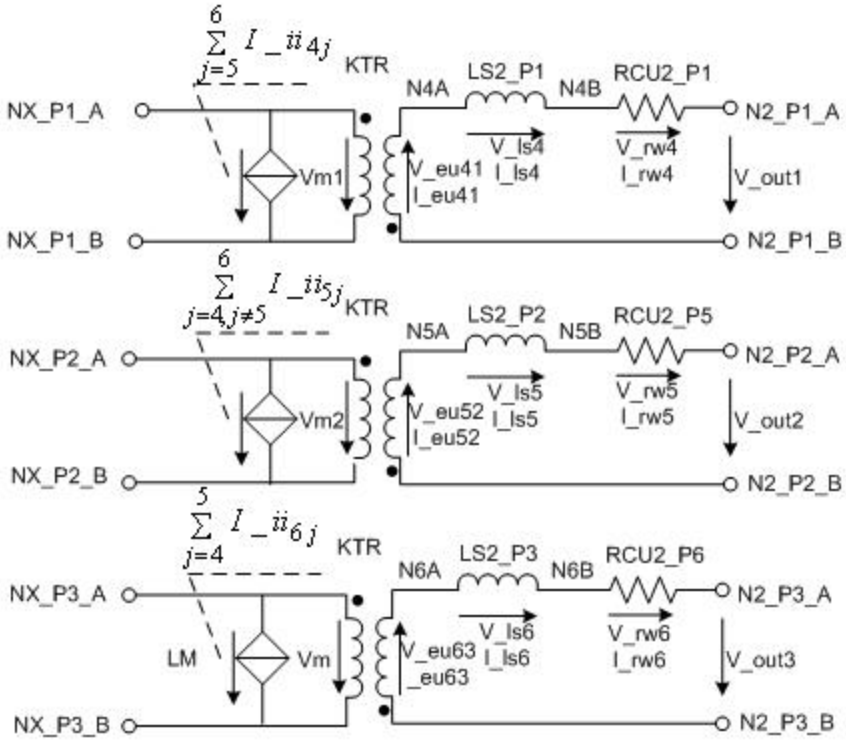


Figure 2. Equivalent circuit of the linear six-winding transformer secondary side

Note: The terms primary and secondary are not intended in the strict sense. It simply means to combine exactly one macro of the primary side with at least one macro of the secondary side.

For each coil make sure a network path exists to the ground node. Within the dialog users can define the main and leakage inductances, the equivalent resistances for iron losses, the winding resistances for each side, and the initial values of the current. To define the parameter values, enter a numerical value, a variable, or expression in the corresponding field of parameter list.

Top

Assumptions and Limitations

This model does not take into account the magnetizing characteristics of the core.

Top

Mathematical Description

Equations that are used to describe the linear six-winding transformer secondary side are as follows:

Table 1

Phase 1	Phase 2	Phase 3
$I\_ii14 = K_{TR} \cdot I\_rw4$	$I\_ii24 = K_{TR23} \cdot I\_rw4$	$I\_ii34 = K_{TR22} \cdot I\_rw4$
$I\_ii15 = K_{TR21} \cdot I\_rw5$	$I\_ii25 = K_{TR} \cdot I\_rw5$	$I\_ii35 = K_{TR21} \cdot I\_rw5$
$I\_ii16 = K_{TR22} \cdot I\_rw6$	$I\_ii26 = K_{TR23} \cdot I\_rw6$	$I\_ii36 = K_{TR} \cdot I\_rw6$
$V\_eu41 = v_{m1} \cdot K_{TR}$	$V\_eu52 = v_{m2} \cdot K_{TR}$	$V\_eu63 = v_{m3} \cdot K_{TR}$
$P_{si}\_ls4 = I\_ls4 \cdot L_{ls2}\_p1$	$P_{si}\_ls5 = I\_ls5 \cdot L_{ls2}\_p2$	$P_{si}\_ls6 = I\_ls6 \cdot L_{ls2}\_p3$
$V\_ls4 = \frac{d\ P_{si}\_ls4}{d\ t}$	$V\_ls5 = \frac{d\ P_{si}\_ls5}{d\ t}$	$V\_ls6 = \frac{d\ P_{si}\_ls6}{d\ t}$
$V\_rw4 = I\_rw4 \cdot R_{cu2}\_p1$	$V\_rw5 = I\_rw5 \cdot R_{cu2}\_p2$	$V\_rw6 = I\_rw6 \cdot R_{cu2}\_p3$
$I\_ii14 = K_{TR} \cdot I\_rw4$	$I\_ii24 = K_{TR23} \cdot I\_rw4$	$I\_ii34 = K_{TR22} \cdot I\_rw4$

Where K_{TR21}=0.5*K_{TR}, K_{TR22}=0.25*K_{TR}, K_{TR23}=0.75*K_{TR}, for small-power transformers, and K_{TR21}=0.5*K_{TR}, K_{TR22}=0.333*K_{TR}, K_{TR23}=0.667*K_{TR}, for large-power transformers

Top

Netlist Syntax

COUPL tfr3p2 ?InstanceName(@InstanceName):(@(Refbase)@(ID)) nx_p1_a := %0 , nx_p1_b := %1 , nx_p2_a := %2 , nx_p2_b := %3 , nx_p3_a := %4 , nx_p3_b := %5 , n2_p1_a := %6 , n2_p1_b := %7 , n2_p2_a := %8 , n2_p2_b := %9 , n2_p3_a := %10 , n2_p3_b := %11 (ls2_p1 := @ls2_p1 , ls2_p2 := @ls2_p2 , ls2_p3 := @ls2_p3 , ktr0 := @ktr0 , ktr1 := @ktr1 , ktr2 := @ktr2 , ktr3 := @ktr3 , i0_ls4 := @i0_ls4 , i0_ls5 := @i0_ls5 , i0_ls6 := @i0_ls6 , rcu2_p1 := @rcu2_p1 , rcu2_p2 := @rcu2_p2 , rcu2_p3 := @rcu2_p3) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 2

Name	Port/Terminal Description	Nature/Data Type
N1_P1_A	Positive node on the primary side of phase 1	electrical
N1_P1_B	Negative node on the primary side of phase 1	electrical
N2_P1_A	Positive node on the secondary side of phase 1	electrical
N2_P1_B	Negative node on the secondary side of phase 1	electrical
N1_P2_A	Positive node on the primary side of phase 2	electrical
N1_P2_B	Negative node on the primary side of phase 2	electrical
N2_P2_A	Positive node on the secondary side of phase 2	electrical
N2_P2_B	Negative node on the secondary side of phase 2	electrical
N1_P3_A	Positive node on the primary side of phase 3	electrical
N1_P3_B	Negative node on the primary side of phase 3	electrical
N2_P3_A	Positive node on the secondary side of phase 3	electrical
N2_P3_B	Negative node on the secondary side of phase 3	electrical

Top

Parameters

Table 3

Name	Description	Data Type	Default Value[Unit]
LS2_P1	Leakage Inductance Secondary Side of Phase 1	real	0.001 [H]
LS2_P2	Leakage Inductance Secondary Side of Phase 2	real	0.001 [H]
LS2_P3	Leakage Inductance Secondary Side of Phase 3	real	0.001 [H]
RCU2_P1	Winding Resistance Secondary Side of Phase 1	real	1.0e-6 [Ohm]
RCU2_P2	Winding Resistance Secondary Side of Phase 2	real	1.0e-6 [Ohm]
RCU2_P3	Winding Resistance Secondary Side of Phase 3	real	1.0e-6 [Ohm]
KTR	Winding Ratio	real	1.0
KTR1	Coupling factor middle to outer limb	real	0.5
KTR2	Coupling factor outer to outer limb	real	0.75
KTR3	Coupling factor outer to middle limb	real	0.25
I0_IS4	Initial Current Secondary Side of Phase 1	real	0 [A]
I0_IS5	Initial Current Secondary Side of Phase 2	real	0 [A]
I0_IS6	Initial Current Secondary Side of Phase 3	real	0 [A]

Top

Example

See Three-Phase Six-Winding Transformer Example

Top

References

Six-Winding Transformer Large-Power

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

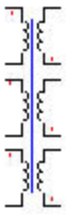


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents linear large-power six-winding transformer based on the equivalent circuit shown in Figure 2, and the structure of the iron core as shown in Figure 3. For each coil, make sure a network path exists to the ground node. Within the dialog box, you can define the main and leakage inductances, the equivalent resistances for iron losses, the winding resistances for each side, and the initial values of the current. To define the parameter values, enter a numerical value, a variable, or expression in the corresponding field of parameter list. The six-winding large-power and small-power transformers differ in the core structure, as described in detail in the Mathematical Description of the Model section.

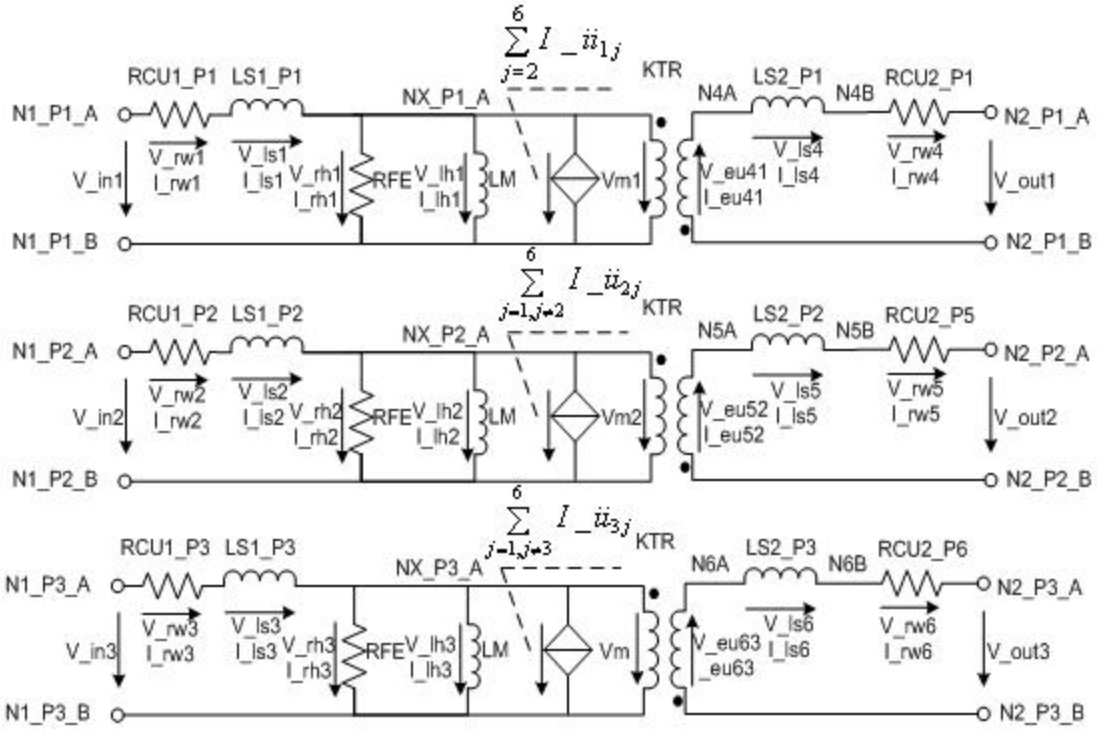


Figure 2. Equivalent circuit of the Linear Six-Winding Transformer

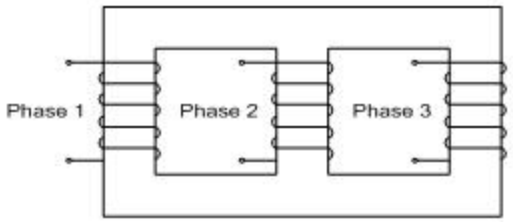


Figure 3. Structure of the iron core of the six-winding transformer(large power)

Top

Assumptions and Limitations

This model does not take into account the magnetizing characteristics of the core.

Top

Mathematical Description

Table 1

	Phase 1	Phase 2	Phase 3
Primary Side	$V_{\_rw1} = I_{\_rw1} \cdot R_{cu1\_p1}$	$V_{\_rw2} = I_{\_rw2} \cdot R_{cu1\_p2}$	$V_{\_rw1} = I_{\_rw1} \cdot R_{cu1\_p3}$
	$P_{si\_ls1} = I_{\_ls1} \cdot L_{ls1\_p1}$	$P_{si\_ls2} = I_{\_ls2} \cdot L_{ls1\_p2}$	$P_{si\_ls5} = I_{\_ls5} \cdot L_{ls1\_p3}$
	$V_{\_ls1} = \frac{d\ P_{si\_ls1}}{d\ t}$	$V_{\_ls2} = \frac{d\ P_{si\_ls2}}{d\ t}$	$V_{\_ls5} = \frac{d\ P_{si\_ls5}}{d\ t}$
	$P_{si\_lh1} = I_{\_lh1} \cdot L_M$	$P_{si\_lh2} = I_{\_lh2} \cdot L_M$	$P_{si\_lh3} = I_{\_lh3} \cdot L_M$
	$V_{\_lh1} = \frac{d\ P_{si\_lh1}}{d\ t}$	$V_{\_lh2} = \frac{d\ P_{si\_lh2}}{d\ t}$	$V_{\_lh3} = \frac{d\ P_{si\_lh3}}{d\ t}$
	$V_{\_rh1} = I_{\_rh1} \cdot R_{FE}$	$V_{\_rh2} = I_{\_rh2} \cdot R_{FE}$	$V_{\_rh3} = I_{\_rh3} \cdot R_{FE}$
	$I_{\_ii12} = K_{TR11} \cdot I_{\_rw2}$	$I_{\_ii21} = K_{TR13} \cdot I_{\_rw1}$	$I_{\_ii31} = K_{TR12} \cdot I_{\_rw1}$
	$I_{\_ii12} = K_{TR12} \cdot I_{\_rw3}$	$I_{\_ii23} = K_{TR13} \cdot I_{\_rw3}$	$I_{\_ii32} = K_{TR11} \cdot I_{\_rw2}$
	$I_{\_ii14} = K_{TR} \cdot I_{\_rw4}$	$I_{\_ii24} = K_{TR23} \cdot I_{\_rw4}$	$I_{\_ii34} = K_{TR22} \cdot I_{\_rw4}$
	$I_{\_ii15} = K_{TR21} \cdot I_{\_rw5}$	$I_{\_ii25} = K_{TR} \cdot I_{\_rw5}$	$I_{\_ii35} = K_{TR21} \cdot I_{\_rw5}$
	$I_{\_ii16} = K_{TR22} \cdot I_{\_rw6}$	$I_{\_ii26} = K_{TR23} \cdot I_{\_rw6}$	$I_{\_ii36} = K_{TR} \cdot I_{\_rw6}$
Secondary Side	$V_{\_eu41} = v_{m1} \cdot K_{TR}$	$V_{\_eu52} = v_{m2} \cdot K_{TR}$	$V_{\_eu63} = v_{m3} \cdot K_{TR}$
	$P_{si\_ls4} = I_{\_ls4} \cdot L_{ls2\_p1}$	$P_{si\_ls5} = I_{\_ls5} \cdot L_{ls2\_p2}$	$P_{si\_ls6} = I_{\_ls6} \cdot L_{ls2\_p3}$
	$V_{\_ls4} = \frac{d\ P_{si\_ls4}}{d\ t}$	$V_{\_ls5} = \frac{d\ P_{si\_ls5}}{d\ t}$	$V_{\_ls6} = \frac{d\ P_{si\_ls6}}{d\ t}$
	$V_{\_rw4} = I_{\_rw4} \cdot R_{cu2\_p1}$	$V_{\_rw5} = I_{\_rw5} \cdot R_{cu2\_p2}$	$V_{\_rw6} = I_{\_rw6} \cdot R_{cu2\_p3}$

Where K_{TR11}=0.5, K_{TR12}=0.333, K_{TR13}=0.667, and
K_{TR21}=0.5*KTR, K_{TR22}=0.333*KTR, K_{TR23}=0.667*KTR for large power transformers

Top

Netlist Syntax

COUPL tfr3p6wl ?InstanceName(@InstanceName):(@(Refbase)@(ID)) n1_p1_a := %0 , n1_ p1_b := %1 , n1_p2_a := %2 , n1_p2_b := %3 , n1_p3_a := %4 , n1_p3_b := %5 , n2_p1_a := %6 , n2_p1_b := %7 , n2_p2_a := %8 , n2_p2_b := %9 , n2_p3_a := %10 , n2_p3_b := %11 (ls1_p1 := @ls1_p1 , ls1_p2 := @ls1_p2 , ls1_p3 := @ls1_p3 , ls2_p1 := @ls2_p1 , ls2_p2 := @ls2_p2 , ls2_p3 := @ls2_p3 , rfe := @rfe , ktr := @ktr , i0_ls1 := @i0_ls1 , i0_ls2 := @i0_ls2 , i0_ls3 := @i0_ls3 , i0_ls4 := @i0_ls4 , i0_ls5 := @i0_ls5 , i0_ls6 := @i0_ls6 , lm := @lm , rcu1_p1 := @rcu1_p1 , rcu1_p2 := @rcu1_p2 , rcu1_p3 := @rcu1_p3 , rcu2_p1 := @rcu2_p1 , rcu2_p2 := @rcu2_p2 , rcu2_p3 := @rcu2_p3) DST: SIM(Type:=SimVHDL) SRC: DB(File:- :=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 2

Name	Port/Terminal Description	Nature/Data Type
N1_P1_A	Positive node on the primary side of phase 1	electrical
N1_P1_B	Negative node on the primary side of phase 1	electrical
N2_P1_A	Positive node on the secondary side of phase 1	electrical
N2_P1_B	Negative node on the secondary side of phase 1	electrical
N1_P2_A	Positive node on the primary side of phase 2	electrical
N1_P2_B	Negative node on the primary side of phase 2	electrical
N2_P2_A	Positive node on the secondary side of phase 2	electrical
N2_P2_B	Negative node on the secondary side of phase 2	electrical
N1_P3_A	Positive node on the primary side of phase 3	electrical
N1_P3_B	Negative node on the primary side of phase 3	electrical
N2_P3_A	Positive node on the secondary side of phase 3	electrical
N2_P3_B	Negative node on the secondary side of phase 3	electrical

Top

Parameters

Table 3

Name	Description	Data Type	Default Value[Unit]
LS1_P1	Leakage Inductance Primary Side of Phase 1	real	0.001 [H]
LS2_P1	Leakage Inductance Secondary Side of Phase 1	real	0.001 [H]
LS1_P2	Leakage Inductance Primary Side of Phase 2	real	0.001 [H]
LS2_P2	Leakage Inductance Secondary Side of Phase 2	real	0.001 [H]
LS1_P3	Leakage Inductance Primary Side of Phase 3	real	0.001 [H]
LS2_P3	Leakage Inductance Secondary Side of Phase 3	real	0.001 [H]
RCU1_P1	Winding Resistance Primary Side of Phase 1	real	1.0e-6 [Ohm]
RCU2_P1	Winding Resistance Secondary Side of Phase 1	real	1.0e-6 [Ohm]
RCU1_P2	Winding Resistance Primary Side of Phase 2	real	1.0e-6 [Ohm]
RCU2_P2	Winding Resistance Secondary Side of Phase 2	real	1.0e-6 [Ohm]
RCU1_P3	Winding Resistance Primary Side of Phase 3	real	1.0e-6 [Ohm]
RCU2_P3	Winding Resistance Secondary Side of Phase 3	real	1.0e-6 [Ohm]
LM	Magnetizing Inductance	real	0.1 [H]
RFE	Equivalent Resistance for Iron Losses	real	1.oe18 [Ohm]
KTR	Winding Ratio	real	1.0
I0_IS1	Initial Current Primary Side of Phase 1	real	0 [A]
I0_IS2	Initial Current Primary Side of Phase 2	real	0 [A]
I0_IS3	Initial Current Primary Side of Phase 3	real	0 [A]
I0_IS4	Initial Current Secondary Side of Phase 1	real	0 [A]
I0_IS5	Initial Current Secondary Side of Phase 2	real	0 [A]
I0_IS6	Initial Current Secondary Side of Phase 3	real	0 [A]

Top

Example

This example demonstrates the setup of a Three-Phase transformer. In this example, the transformer is configured for large power (KTR1=0.5, KTR2=0.333, and KTR3=0.667).

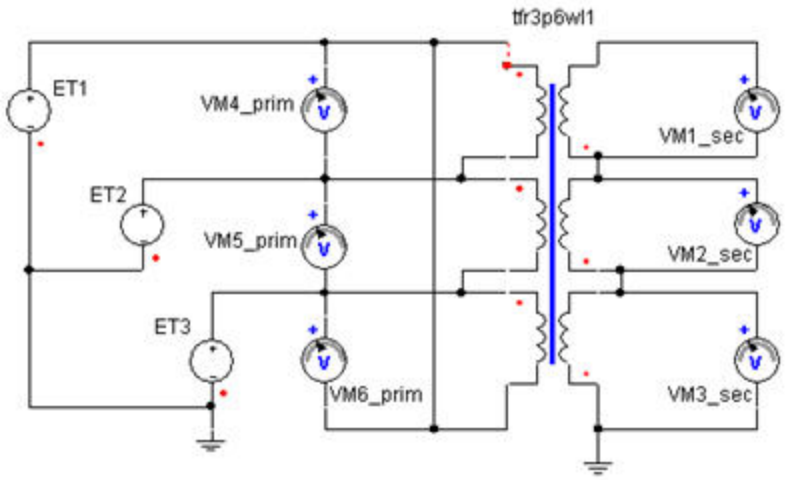


Figure 4. Application examples of the VHDL-AMS Linear Three-Phase Large Power Transformer model

Table 4. System Parameters

Component	Parameter	Value [unit]
Time Controlled Voltage Source (Sinusoidal) ET1/ET2/ET3	AMPL	0.5k [V]
	freq	50 [Hz]
	TPERIO	20m [S]
	PHASE (ET1/ET2/ET3)	0 / -120 / -240 [Deg]
Linear Three-Phase-Six-Winding Transformer (Large Power) tfr3p6wl1	ktr	1
	lm	0.1 [H]

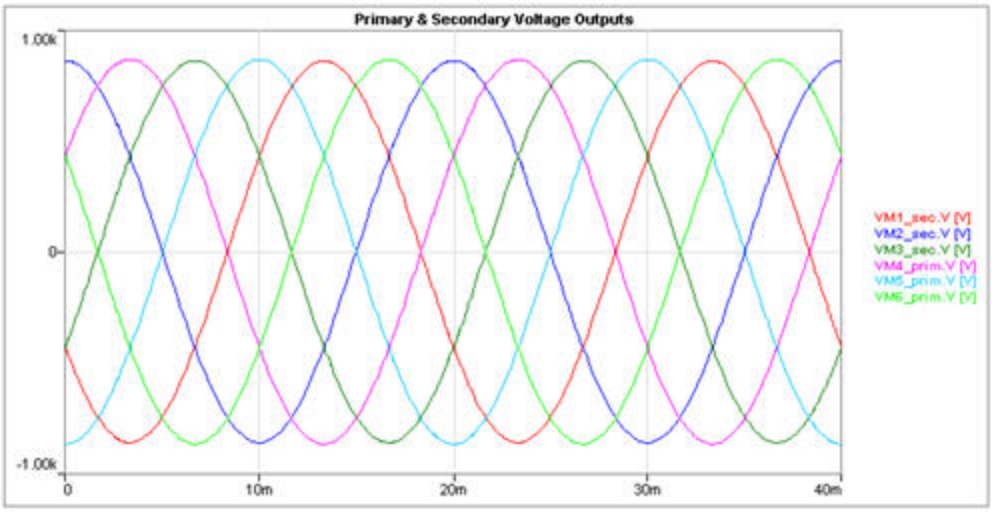


Figure 5. Simulation results-outputs on primary and secondary of transformer.

Top

References

Six-Winding Transformer Small-Power

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

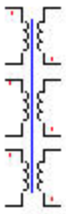


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents linear small-power six-winding transformers based on the equivalent circuit shown in Figure 2, and the structure of the iron core shown in Figure 3. For each coil, make sure a network path exists to the ground node. Within the dialog box, you can define the main and leakage inductances, the equivalent resistances for iron losses, the winding resistances for each side, and the initial values of the current. To define the parameter values, enter a numerical value, a variable, or expression in the corresponding field of parameter list. The six-winding large-power and small-power transformers differ in the core structure, as described in detail in the Mathematical Description of the Model section.

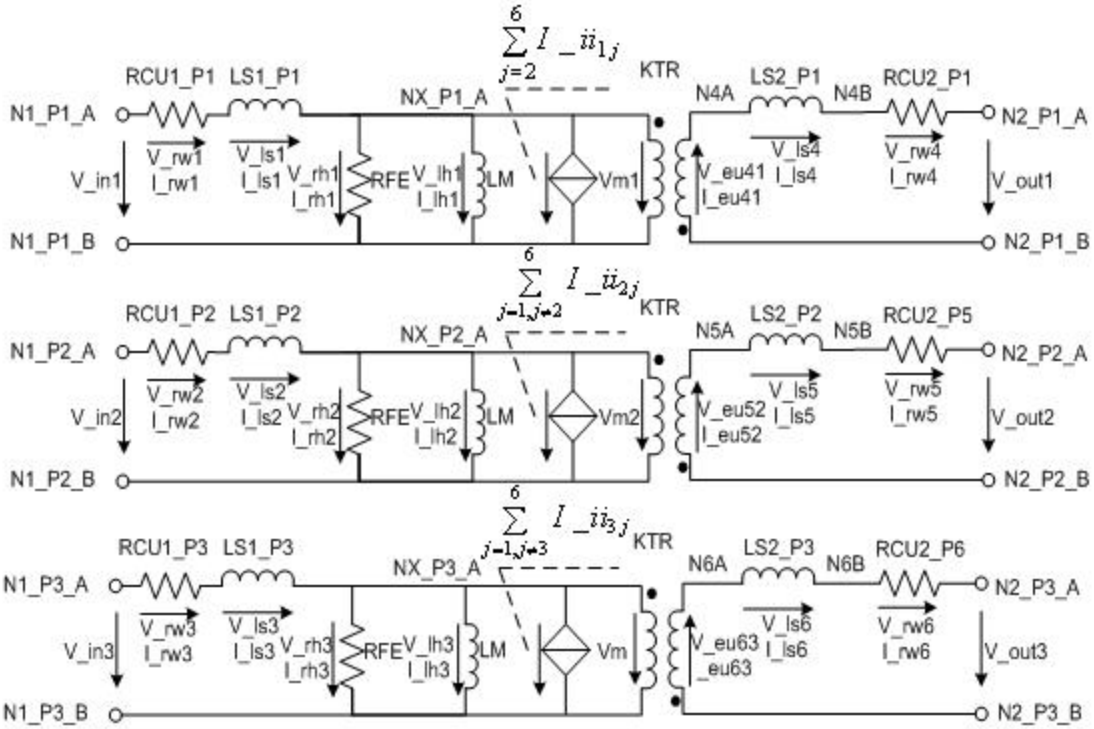


Figure 2. Equivalent circuit of the Linear Six-Winding transformer

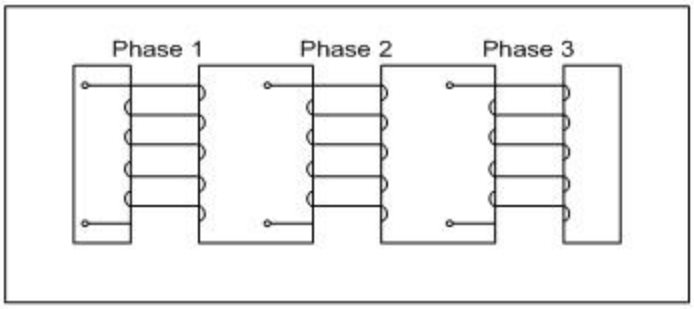


Figure 3. Structure of the iron core of the six-winding transformer(small power)

Top

Assumptions and Limitations

This model does not take into account the magnetizing characteristics of the core.

Top

Mathematical Description

Equations that are used to describe the linear six-winding transformer are listed as follows:

Table 1

	Phase 1	Phase 2	Phase 3
Primary Side	$V\_rw1 = I\_rw1 \cdot R_{cu1}\_p1$	$V\_rw2 = I\_rw2 \cdot R_{cu1}\_p2$	$V\_rw1 = I\_rw1 \cdot R_{cu1}\_p3$
	$P_{si}\_ls1 = I\_ls1 \cdot L_{ls1}\_p1$	$P_{si}\_ls2 = I\_ls2 \cdot L_{ls1}\_p2$	$P_{si}\_ls5 = I\_ls5 \cdot L_{ls1}\_p3$
	$V\_ls1 = \frac{d\ P_{si}\_ls1}{d\ t}$	$V\_ls2 = \frac{d\ P_{si}\_ls2}{d\ t}$	$V\_ls5 = \frac{d\ P_{si}\_ls5}{d\ t}$
	$P_{si}\_lh1 = I\_lh1 \cdot L_M$	$P_{si}\_lh2 = I\_lh2 \cdot L_M$	$P_{si}\_lh3 = I\_lh3 \cdot L_M$
	$V\_lh1 = \frac{d\ P_{si}\_lh1}{d\ t}$	$V\_lh2 = \frac{d\ P_{si}\_lh2}{d\ t}$	$V\_lh3 = \frac{d\ P_{si}\_lh3}{d\ t}$
	$V\_rh1 = I\_rh1 \cdot R_{FE}$	$V\_rh2 = I\_rh2 \cdot R_{FE}$	$V\_rh3 = I\_rh3 \cdot R_{FE}$
	$I\_ii12 = K_{TR11} \cdot I\_rw2$	$I\_ii21 = K_{TR13} \cdot I\_rw1$	$I\_ii31 = K_{TR12} \cdot I\_rw1$
	$I\_ii12 = K_{TR12} \cdot I\_rw3$	$I\_ii23 = K_{TR13} \cdot I\_rw3$	$I\_ii32 = K_{TR11} \cdot I\_rw2$
	$I\_ii14 = K_{TR} \cdot I\_rw4$	$I\_ii24 = K_{TR23} \cdot I\_rw4$	$I\_ii34 = K_{TR22} \cdot I\_rw4$
	$I\_ii15 = K_{TR21} \cdot I\_rw5$	$I\_ii25 = K_{TR} \cdot I\_rw5$	$I\_ii35 = K_{TR21} \cdot I\_rw5$
	$I\_ii16 = K_{TR22} \cdot I\_rw6$	$I\_ii26 = K_{TR23} \cdot I\_rw6$	$I\_ii36 = K_{TR} \cdot I\_rw6$
Secondary Side	$V\_eu41 = v_{m1} \cdot K_{TR}$	$V\_eu52 = v_{m2} \cdot K_{TR}$	$V\_eu63 = v_{m3} \cdot K_{TR}$
	$P_{si}\_ls4 = I\_ls4 \cdot L_{ls2}\_p1$	$P_{si}\_ls5 = I\_ls5 \cdot L_{ls2}\_p2$	$P_{si}\_ls6 = I\_ls6 \cdot L_{ls2}\_p3$
	$V\_ls4 = \frac{d\ P_{si}\_ls4}{d\ t}$	$V\_ls5 = \frac{d\ P_{si}\_ls5}{d\ t}$	$V\_ls6 = \frac{d\ P_{si}\_ls6}{d\ t}$
	$V\_rw4 = I\_rw4 \cdot R_{cu2}\_p1$	$V\_rw5 = I\_rw5 \cdot R_{cu2}\_p2$	$V\_rw6 = I\_rw6 \cdot R_{cu2}\_p3$

Where K_{TR11}=0.5, K_{TR12}=0.25, K_{TR13}=0.75, and
K_{TR21}=0.5*KTR, K_{TR22}=0.25*KTR, K_{TR23}=0.75*KTR for small power transformers

Top

Netlist Syntax

COUPL tfr3p6ws ?InstanceName(@InstanceName):(@(Refbase)@(ID)) n1_p1_a := %0 , n1_p1_b := %1 , n1_p2_a := %2 , n1_p2_b := %3 , n1_p3_a := %4 , n1_p3_b := %5 , n2_p1_a := %6 , n2_p1_b := %7 , n2_p2_a := %8 , n2_p2_b := %9 , n2_p3_a := %10 , n2_p3_b := %11 (ls1_p1 := @ls1_p1 , ls1_p2 := @ls1_p2 , ls1_p3 := @ls1_p3 , ls2_p1 := @ls2_p1 , ls2_p2 := @ls2_p2 , ls2_p3 := @ls2_p3 , rfe := @rfe , ktr := @ktr , i0_ls1 := @i0_ls1 , i0_ls2 := @i0_ls2 , i0_ls3 := @i0_ls3 , i0_ls4 := @i0_ls4 , i0_ls5 := @i0_ls5 , i0_ls6 := @i0_ls6 , lm := @lm , rcu1_p1 := @rcu1_p1 , rcu1_p2 := @rcu1_p2 , rcu1_p3 := @rcu1_p3 , rcu2_p1 := @rcu2_p1 , rcu2_p2 := @rcu2_p2 , rcu2_p3 := @rcu2_p3) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 2

Name	Port/Terminal Description	Nature/Data Type
N1_P1_A	Positive node on the primary side of phase 1	electrical
N1_P1_B	Negative node on the primary side of phase 1	electrical
N2_P1_A	Positive node on the secondary side of phase 1	electrical
N2_P1_B	Negative node on the secondary side of phase 1	electrical
N1_P2_A	Positive node on the primary side of phase 2	electrical
N1_P2_B	Negative node on the primary side of phase 2	electrical
N2_P2_A	Positive node on the secondary side of phase 2	electrical
N2_P2_B	Negative node on the secondary side of phase 2	electrical
N1_P3_A	Positive node on the primary side of phase 3	electrical
N1_P3_B	Negative node on the primary side of phase 3	electrical
N2_P3_A	Positive node on the secondary side of phase 3	electrical
N2_P3_B	Negative node on the secondary side of phase 3	electrical

Top

Parameters

Table 3

Name	Description	Data Type	Default Value[Unit]
LS1_P1	Leakage Inductance Primary Side of Phase 1	real	0.001 [H]
LS2_P1	Leakage Inductance Secondary Side of Phase 1	real	0.001 [H]
LS1_P2	Leakage Inductance Primary Side of Phase 2	real	0.001 [H]
LS2_P2	Leakage Inductance Secondary Side of Phase 2	real	0.001 [H]
LS1_P3	Leakage Inductance Primary Side of Phase 3	real	0.001 [H]
LS2_P3	Leakage Inductance Secondary Side of Phase 3	real	0.001 [H]
RCU1_ P1	Winding Resistance Primary Side of Phase 1	real	1.0e-6 [Ohm]
RCU2_ P1	Winding Resistance Secondary Side of Phase 1	real	1.0e-6 [Ohm]
RCU1_ P2	Winding Resistance Primary Side of Phase 2	real	1.0e-6 [Ohm]
RCU2_ P2	Winding Resistance Secondary Side of Phase 2	real	1.0e-6 [Ohm]
RCU1_ P3	Winding Resistance Primary Side of Phase 3	real	1.0e-6 [Ohm]
RCU2_ P3	Winding Resistance Secondary Side of Phase 3	real	1.0e-6 [Ohm]
LM	Magnetizing Inductance	real	0.1 [H]
RFE	Equivalent Resistance for Iron Losses	real	1.oe18 [Ohm]
KTR	Winding Ratio	real	1.0
I0_IS1	Initial Current Primary Side of Phase 1	real	0 [A]
I0_IS2	Initial Current Primary Side of Phase 2	real	0 [A]
I0_IS3	Initial Current Primary Side of Phase 3	real	0 [A]
I0_IS4	Initial Current Secondary Side of Phase 1	real	0 [A]
I0_IS5	Initial Current Secondary Side of Phase 2	real	0 [A]
I0_IS6	Initial Current Secondary Side of Phase 3	real	0 [A]

Top

Example

This example demonstrates the setup of a Three-Phase transformer. In this example, the transformer is configured for large power (KTR1=0.5, KTR2=0.75, and KTR3=0.25).

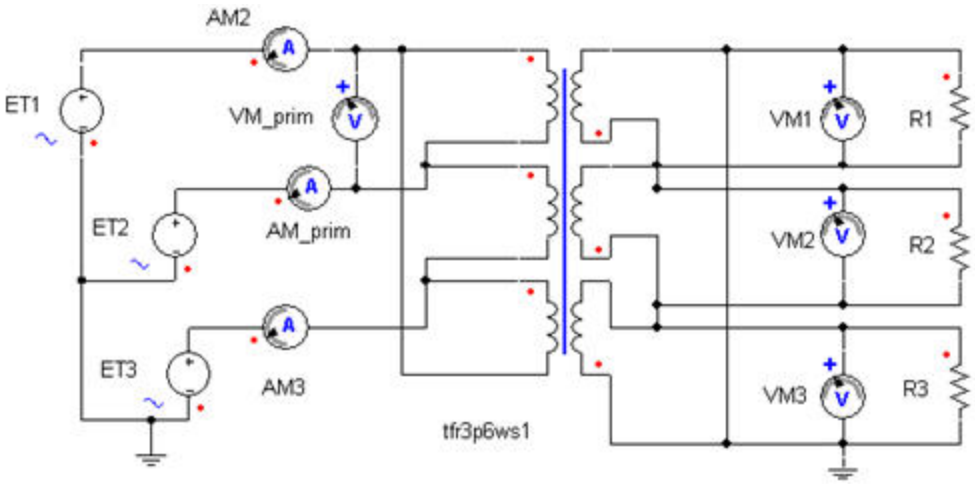


Figure 4. Application examples of the VHDL-AMS Linear Three-Phase Large Power Transformer model

Table 4. System Parameters

Component	Parameter	Value [unit]
Time Controlled Voltage Source (Sinusoidal) ET1/ET2/ET3	AMPL	0.326k [V]
	freq	50 [Hz]
	TPERIO	20m [S]
	PHASE (ET1/ET2/ET3)	0 / -120 / -240 [Deg]
Linear Three-Phase-Six-Winding Transformer (Small Power) tfr3p6ws1	ktr	2
	lm	0.1 [H]

Resistor r1/r2/r3	r	10 [Ohm]

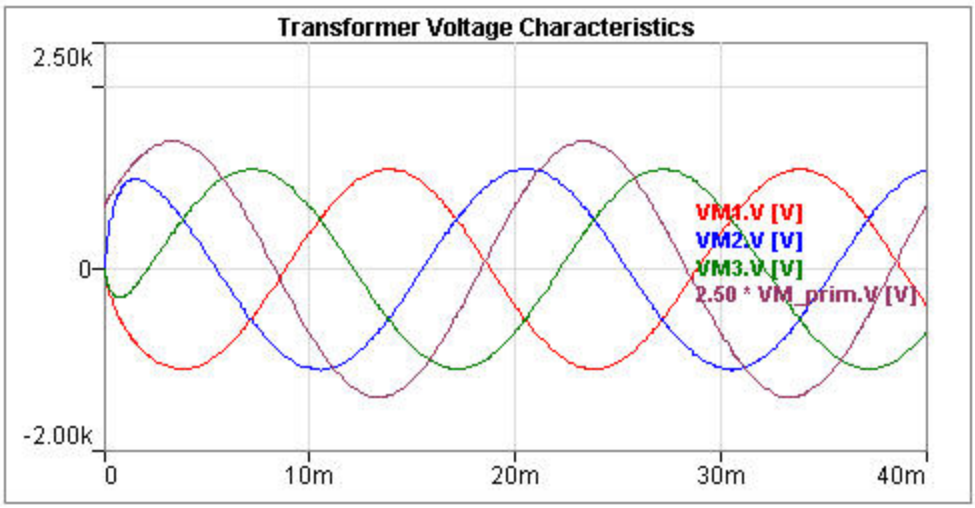


Figure 5. Simulation results-voltage characteristics.

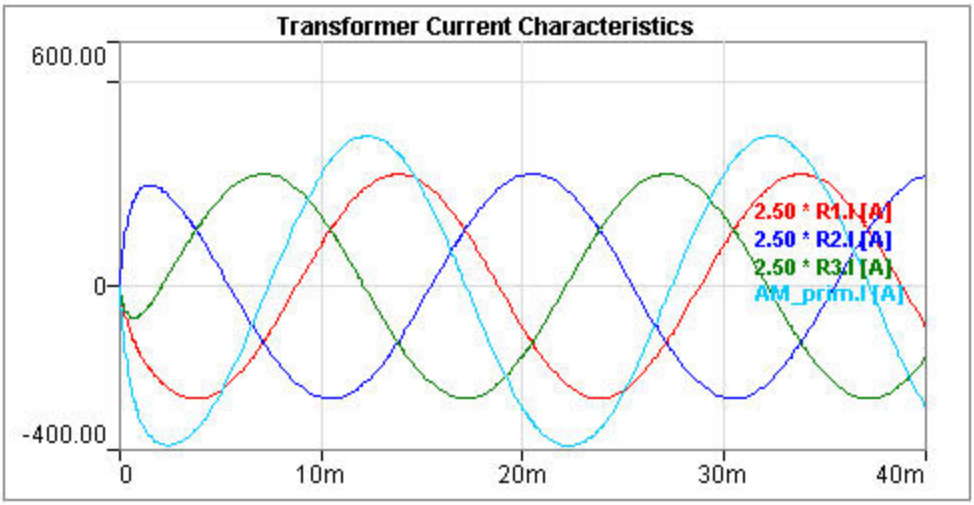


Figure 6. Simulation results-current characteristics.

Top

References

Three-Phase Six-Winding Transformer Example

This example demonstrates the use of Primary and Secondary side Three-Phase Six-Winding Transformer models. The primary side model is driven by a three-phase source. Two secondary sides are modeled, one connected to a resistive load in a star configuration, the other connected in a delta configuration.

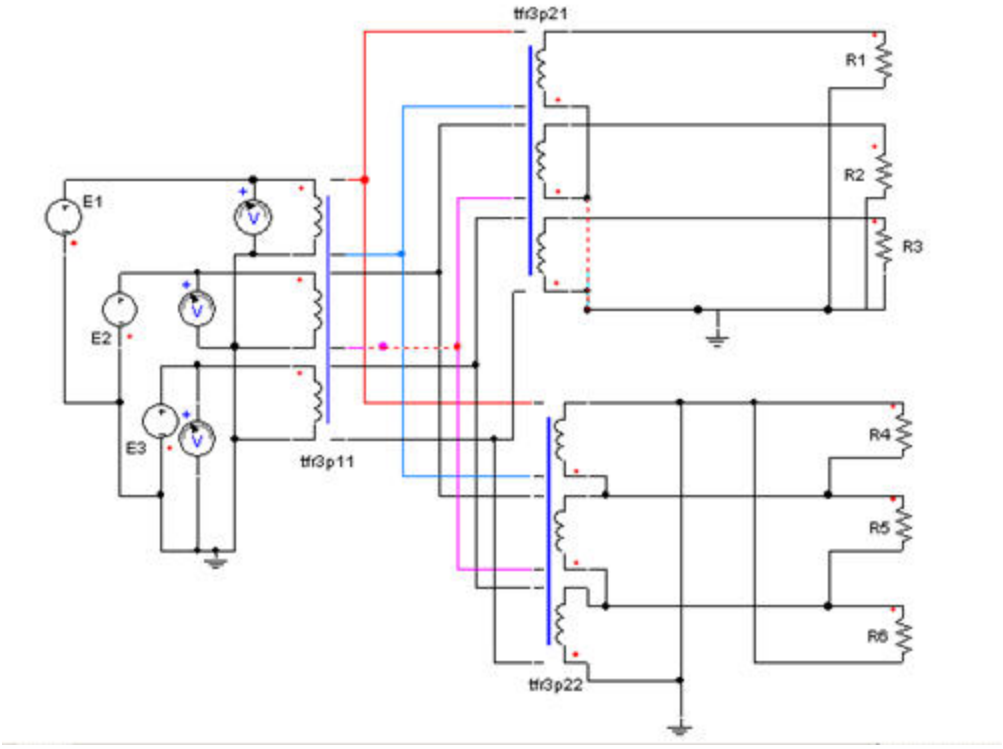


Figure 1. Application example of the VHDL-AMS Three-Phase Six-Winding Primary and Secondary Transformer Models

Table 1. System Parameters

Component	Parameter	Value [unit]
Time Controlled Voltage Source (Sinusoidal) ET1/ET2/ET3	AMPL	0.326k [V]
	freq	50 [Hz]
	TPERIO	20m [S]
	PHASE (ET1/ET2/ET3)	0 / -120 / -240 [Deg]

Transformer Primary Side tfr3p11	ktr1/ktr2/ktr3	0.5/0.25/0.75
Transformer Secondary Side tfr3p21	ktr0	1.0
	ktr1/ktr2/ktr3	0.5/0.25/0.75
Transformer Secondary Side tfr3p22	ktr0	1.75
	ktr1/ktr2/ktr3	0.5/0.25/0.75
Resistor r1/r2/r3/r4/r5/r6	r	1k [Ohm]

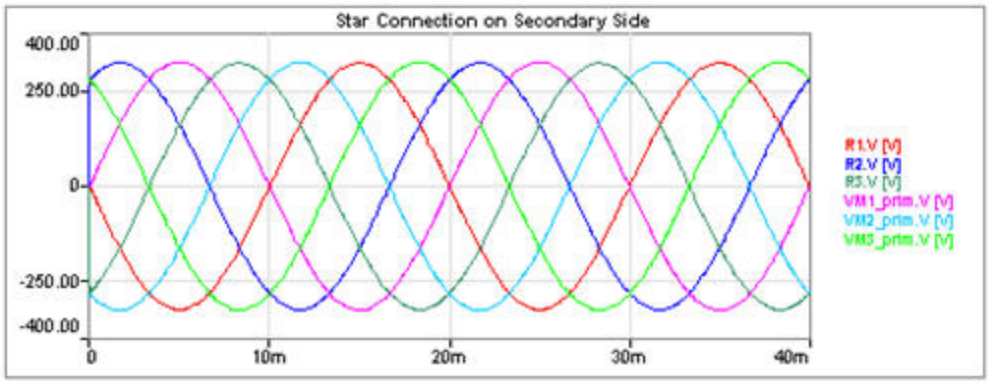


Figure 2. Simulation results-primary and secondary side voltages for star configured load (R1/R2/R3).

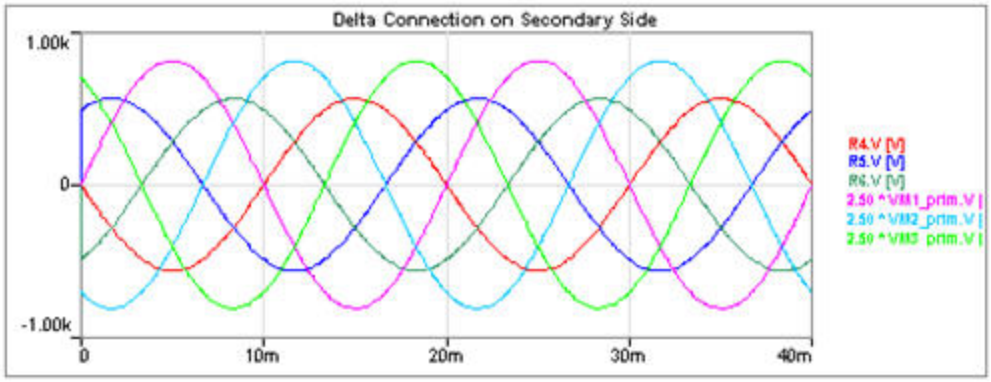


Figure 3. Simulation results-primary and secondary side voltages for delta configured load (R4/R5/R6).

Using Measuring Instruments

Twin Builder provides measuring instruments for different physical domains. These meters provide through and across quantities as well as power for every physical domain used in a simulation model. All meters are ideal components, that means, no relevant quantities are assumed to be '0'.

Meters have no user-defined parameters. They have at least one conservative node and are connected with a specific physical domain circuit.

The meters can provide the measured value at a special output pin, which can be connected with input parameters in the Schematic.

Measuring instruments can be used for modeling in several ways.

• For physical domain components, measuring instruments can provide through, across, and power quantities for controlled sources and components.

• For blocks, measuring instruments can provide setpoints, disturbances and reference values.

• For state graphs, measuring instruments can provide values to define comparison functions and synchronization signals.

Meters for following domains are available:

• Electrical

• Fluidic

• Magnetic

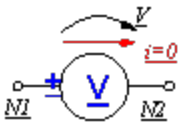
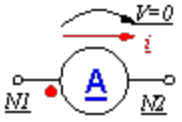
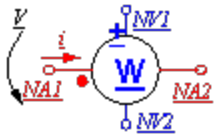
• Mechanical

• Thermal

Measuring components appear in the **Basic Elements VHDLAMS library**. Select the folder *Measurement*, open one of the physical domain folders, and choose a meter for the corresponding quantity.

Electrical Meters

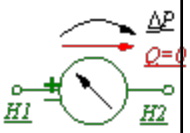
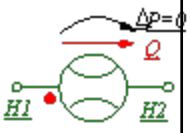
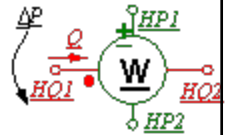
Electrical meters provide voltage (across quantity), current (through quantity), and power for electrical domain components.

Name/Equation	Symbol and Nodes	Outputs	Netlist
Voltmeter (vm) i(t) = 0, R_i = ∞		V [V]...Voltage	COUPL vm ?InstanceName(@InstanceName):(@(Refbase)@(ID)) p := %0 , m := %1 ( ) DST: SIM(Type:-:=SimVHDL) SRC: DB (File:-=@ModelLibraryName, Lang:=VHDLA, Lvl:-:=\\"@Architecture\\"); ;
Ammeter (am) v(t) = 0, R_i = 0		I [A]...Current	COUPL am ?InstanceName(@InstanceName):(@(Refbase)@(ID)) p := %0 , m := %1 ( ) DST: SIM(Type:-:=SimVHDL) SRC: DB (File:-=@ModelLibraryName, Lang:=VHDLA, Lvl:-:=\\"@Architecture\\"); ;
Wattmeter (wm) power(t) = v(t) * i(t)		P [W]...Power	COUPL wm ?InstanceName(@InstanceName):(@(Refbase)@(ID)) pi := %0 , mi := %1 , pv := %2 , mv := %3 ( ) DST: SIM(Type:-

			=SimVHDL) SRC: DB (File:-=@ModelLibraryName, Lang:=VHDLA, Lvl:-:=\\"@Architecture\\"); ;

Fluidic Meters

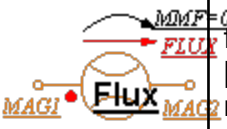
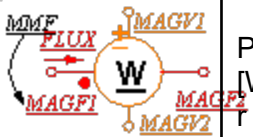
Fluidic meters provide pressure (across quantity), flow rate (through quantity), and power for fluidic domain components.

Name/Equation	Symbol and Nodes	Outputs	Netlist
Manometer (pm) q(t) = 0		P [Pa]...Differential Pressure	COUPL pm ?InstanceName(@InstanceName):(@(Refbase)@(ID)) h1 := %0 , h2 := %1 ( ) DST: SIM(Type:-:=SimVHDL) SRC: DB(File:-=@ModelLibraryName, Lang:=VHDLA, Lvl:-=\\"@A-Architecture\\"); ;
Flow Meter (qm) p(t) = 0		Q [m^3/s]...Flow Rate	COUPL qm ?InstanceName(@InstanceName):(@(Refbase)@(ID)) h1 := %0 , h2 := %1 ( ) DST: SIM(Type:-:=SimVHDL) SRC: DB(File:-=@ModelLibraryName, Lang:=VHDLA, Lvl:-=\\"@A-Architecture\\"); ;
Wattmeter (wm_hyd) power(t) = q(t) * p(t)		P [W]...Power	COUPL wm_hyd ?InstanceName(@InstanceName):(@(Refbase)@(ID)) hp1 := %0 , hp2 := %1 , hq1 := %2 , hq2 := %3 ( ) DST: SIM

			(Type:=SimVHDL) SRC: DB(File:-=@ModelLibraryName, Lang:=VHDLA, Lvl:-=\\"@A-Architecture\\"); ;

Magnetic Meters

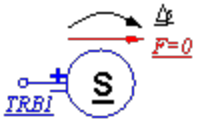
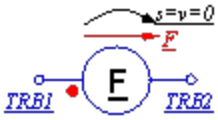
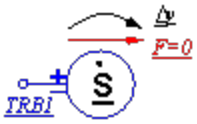
Magnetic meters provide magnetic voltage (across quantity), magnetic flux (through quantity), and power for magnetic domain components.

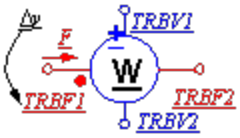
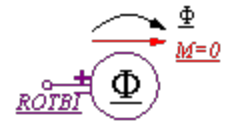
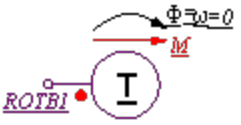
Name/Equation	Symbol and Nodes	Outputs	Netlist
Magnetic Voltmeter (vm_mag) flux(t) = 0		mmf [A]...Magnetic Voltage	COUPL vm_mag ?InstanceName(@InstanceName):(@(Refbase)@(ID)) mag1 := %0 , mag2 := %1 ( ) DST: SIM (Type:=SimVHDL) SRC: DB(File:-=@ModelLibraryName, Lang:=VHDLA, Lvl:-=\\"@A-Architecture\\"); ;
Flux Sensor (fluxm) mmf(t) = 0		flux [W]...Magnetic Flux	COUPL fluxm ?InstanceName(@InstanceName):(@(Refbase)@(ID)) mag1 := %0 , mag2 := %1 ( ) DST: SIM (Type:=SimVHDL) SRC: DB(File:-=@ModelLibraryName, Lang:=VHDLA, Lvl:-=\\"@A-Architecture\\"); ;
Wattmeter (wm_mag) power(t) = flux(t) * mmf(t)		P [W]...Power	COUPL wm_mag ?InstanceName(@InstanceName):(@(Refbase)@(ID)) magv1 := %0 , magv2 := %1 , magf1 := %2 , magf2 := %3 ( ) DST:

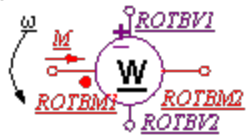
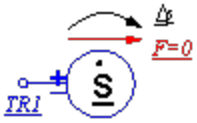
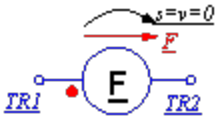
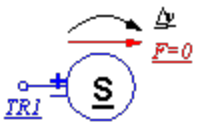
			SIM(Type:- :=SimVHDL) SRC: DB(File:- =@ModelLibraryNa- me, Lang:=VHDLA, Lvl:- =\\"@A- Architecture\\"); ;

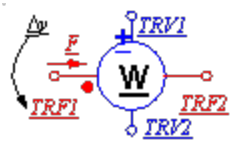
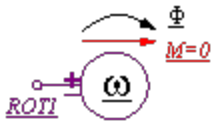
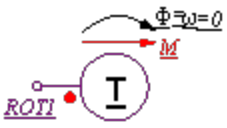
Mechanical Meters

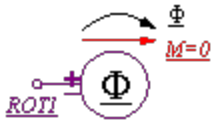
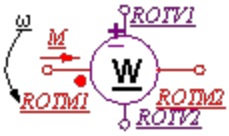
Mechanical meters provide position (across quantity), velocity (across quantity), force (through quantity), and power for mechanical domain components. There are meters for Displacement-Force and Velocity-Force representation. Each representation form has components for translational and rotational systems.

Displacement-Force-Representation - Translational			
Name/Equation	Symbol and Nodes	Outputs	Netlist
Position Sensor (sm_trb) f(t) = 0		S [m]...Displacement	COUPL sm_trb ?InstanceName(@InstanceName):(@(Refbase)@(ID)) trb1 := %0 ( ) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;
Force Sensor (fm_trb) s(t) = 0, v(t) = 0		F [N]...Force	COUPL fm_trb ?InstanceName(@InstanceName):(@(Refbase)@(ID)) trb1 := %0 , trb2 := %1 ( ) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;
Velocity Sensor (vm_trb) f(t) = 0		V [m/s]...Velocity	COUPL vm_trb ?InstanceName(@InstanceName):(@(Refbase)@(ID)) trb1 := %0 ( ) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Wattmeter (wm_trb) power(t) = f(t) * v(t)		P [W]...Power	COUPL wm_trb ?InstanceName(@InstanceName):(@(Refbase)@(ID)) trbf1 := %0 , trbf2 := %1 , trbv1 := %2 , trbv2 := %3 ( ) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:-:=\\"@Architecture\\"); ;
Displacement-Force-Representation - Rotational			
Name/Equation	Symbol and Nodes	Outputs	Netlist
Angle Sensor (sm_rotb) m(t) = 0		Phi [rad]...Angle	COUPL sm_rotb ?InstanceName(@InstanceName):(@(Refbase)@(ID)) rotb1 := %0 ( ) DST: SIM(Type:-:=SimVHDL) SRC: DB (File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:-:=\\"@Architecture\\"); ;
Torque Sensor (fm_rotb) phi(t) = 0, omega(t) = 0		M [Nm]...Torque	COUPL fm_rotb ?InstanceName(@InstanceName):(@(Refbase)@(ID)) rotb1 := %0 , rotb2 := %1 ( ) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:-:=\\"@Architecture\\"); ;
Angular Velocity Sensor (vm_rotb) m(t) = 0		OMEGA [rad/s]...Angular Velocity	COUPL vm_rotb ?InstanceName(@InstanceName):(@(Refbase)@(ID)) rotb1 := %0 ( ) DST: SIM(Type:-:=SimVHDL) SRC: DB (File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:-

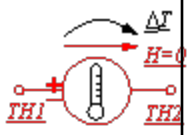
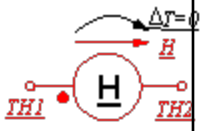
			=\\"@Architecture\\"); ;
Wattmeter (wm_rotb) power(t) = m (t) * omega(t)		P [W]...Power	COUPL wm_rotb ?InstanceName(@InstanceName):(@(Refbase)@(ID)) rotbv1 := %0 , rotbv2 := %1 , rotbm1 := %2 , rotbm2 := %3 ( ) DST: SIM(Type:-:=SimVHDL) SRC: DB (File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:-:=\\"@Architecture\\"); ;
Velocity-Force-Representation - Translational_V			
Name/Equation	Symbol and Nodes	Outputs	Netlist
Velocity Sensor (vm_tr) f(t) = 0		V [m/s]...Velocity	COUPL vm_tr ?InstanceName(@InstanceName):(@(Refbase)@(ID)) tr1 := %0 ( ) DST: SIM(Type:-:=SimVHDL) SRC: DB (File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:-:=\\"@Architecture\\"); ;
Force Sensor (fm_tr) s(t) = 0, v(t) = 0		F [N]...Force	COUPL fm_tr ?InstanceName(@InstanceName):(@(Refbase)@(ID)) tr1 := %0 , tr2 := %1 ( ) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:-:=\\"@Architecture\\"); ;
Position Sensor (sm_tr) f(t) = 0		S [m]...Position	COUPL sm_tr ?InstanceName(@InstanceName):(@(Refbase)@(ID)) tr1 := %0 (s0 := @s0) DST:

			SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:-:=\\"@Architecture\\"); ;
Wattmeter (wm_tr) power(t) = f(t) * v(t)		P [W]...Power	COUPL wm_tr ?InstanceName(@InstanceName):(@(Refbase)@(ID)) trf1 := %0 , trf2 := %1 , trv1 := %2 , trv2 := %3 ( ) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:-:=\\"@Architecture\\"); ;
Velocity-Force-Representation - Rotational_V			
Name/Equation	Symbol and Nodes	Outputs	Netlist
Angular Velocity Sensor (vm_rot) m(t) = 0		omega [rad/s]...Angular Velocity	COUPL vm_rot ?InstanceName(@InstanceName):(@(Refbase)@(ID)) rot1 := %0 ( ) DST: SIM(Type:-:=SimVHDL) SRC: DB (File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:-:=\\"@Architecture\\"); ;
Torque Sensor (fm_rot) phi(t) = 0, omega(t) = 0		M [Nm]...Torque	COUPL fm_rot ?InstanceName(@InstanceName):(@(Refbase)@(ID)) rot1 := %0 , rot2 := %1 ( ) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:-:=\\"@Architecture\\"); ;
Angle Sensor (sm_rot)		Phi [rad]...Angle	COUPL sm_rot ?InstanceName(@In-

m(t) = 0			stanceName):(@(Ref-base)@(ID)) rot1 := %0 (phi0 := @phi0) DST: SIM (Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:-:=\\"@Architecture\\"); ;
Wattmeter (wm_rot) power(t) = m (t) * omega(t)		P [W]...Power	COUPL wm_rot ?In-stanceName(@In-stanceName):(@ (Refbase)@(ID)) rotv1 := %0 , rotv2 := %1 , rotm1 := %2 , rotm2 := %3 ( ) DST: SIM(Type:-:=SimVHDL) SRC: DB (File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:-:=\\"@Architecture\\"); ;

Thermal Meters

Thermal meters provide temperature (across quantity) and heat flow (through quantity) for thermal domain components.

Name/Equation	Symbol and Nodes	Outputs	Netlist
Thermometer (thm) h(t) = 0		T [K]...Temperature	COUPL thm ?InstanceName(@InstanceName):(@(Refbase)@(ID)) th1 := %0 ( ) DST: SIM (Type:=SimVHDL) SRC: DB(File:-=@ModelLibraryName, Lang:=VHDLA, Lvl:-=\\"@A-Architecture\\"); ;
Heat Flow Meter (tfm) T(t) = 0		H [J/s]...Heat Flow	COUPL tfm ?InstanceName(@InstanceName):(@(Refbase)@(ID)) th1 := %0 , th2 := %1 ( ) DST: SIM(Type:-:=SimVHDL) SRC: DB(File:-=@ModelLibraryName, Lang:=VHDLA, Lvl:-=\\"@A-Architecture\\"); ;

Modeling with Physical Domains Components - VHDL-AMS

VHDL-AMS Physical Domains components appear in the *Physical Domains* folder in the **Basic Elements VHDLAMS library**. The folder is subdivided into Fluidic, Magnetic, Mechanical, and Thermal domains. Along with electrical domain components you can model and simulate physical domain systems for different engineering applications. Each domain has its own relation for across and through quantities. In addition, components and wires of a domain are represented by an identical color in the graphical representation of the Schematic.

The Physical Domains folder provides components for the following domains:

- Fluidic Components
- Magnetic Components
- Mechanical Components
- Thermal Components

Note: Mechanical components are subdivided into two categories: Displacement Force Representation, and Velocity Force Representation. Each of these two categories in-turn have two sub-categories: Rotational and Translational.

Fluidic Components - VHDL-AMS

• Hydraulic Capacitance in VHDL-AMS (chyd)

• Hydraulic Inductance in VHDL-AMS (lhyd)

• Pressure Source in VHDL-AMS (p)

• Flow Source in VHDL-AMS (q)

• Linear Orifice in VHDL-AMS (Hydraulic Resistance) (rhyd)

The fluidic domain components appear in the *Fluidic* folder of *Physical Domains* in the **Basic Element VHDLAMS** library. The folder provides fluidic basic components to model simple fluidic equivalent circuits. The fluidic domain uses the pressure as across and the flow rate as through quantity.

By default, components and wires of the fluidic domain are represented in green in the graphical representation of the Schematic. To change the settings of nature colors, choose Tools\Options\Schematic Editor Options, click on the Colors tab, select Wires as the object type, and the color of wires for different domains can be edited. Please note: You can only change the wiring color, not the color of component. symbols.

Across and Through Quantity of the Fluidic Domain

Across	Through	Equation
Pressure [Pa]	Flow Rate [m^2/s^2]	q(t) = K * p(t)

Hydraulic Capacitance - VHDL-AMS

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

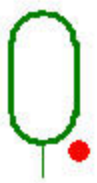


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents a hydraulic resistance that also includes the effects of fluid compressibility, or hydraulic capacitance. The characteristics are based on equations for laminar and turbulent flow through a smooth round pipe supplemented with the differential equation for effect of fluid compressibility.

Top

Assumptions and Limitations

Top

Mathematical Description

The Hydraulic Capacitance model can be described as:

$$q(t) = C_{HYD} \cdot \frac{dp(t)}{dt}$$

$$C_{HYD} = \frac{vol}{b}$$

where C_{HYD} is the equivalent hydraulic capacitance.

Top

Netlist Syntax

COUPL chyd ?InstanceName(@InstanceName):(@(Refbase)@(ID)) h1 := %0 (vol := @vol , b := @b , p0 := @p0) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:-:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
h1	Plus Terminal	fluidic

Top

Parameters

Table 2

Name	Description	Data Type	Default Value[Unit]
vol	Fluid Volume	real	0.0 [m³]
b	Bulk Modulus	real	1.0e9 [N/m²]
p0	Initial Pressure	real	1.0e6 [Pa]

Top

Example

This example illustrates the use of the Hydraulic Capacitance model. The volume if the Hydraulic Capacitance is charged by a pressure source through a hydraulic resistance. The time constant is 1msec.

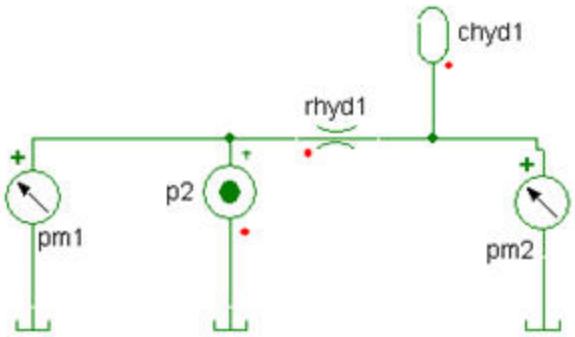


Figure 2. Application examples of the VHDL-AMS Hydraulic Capacitance.

Table 3. System Parameters

Component	Parameter	Value [unit]
Fluidic Pressure Source p2	value	1Meg
Linear Orifice rhyd1	k	1n
Hydraulic Capacitance chyd1	vol	1m [m^3]
	b	1G [N/m^2]
	p0	0 [Pa]

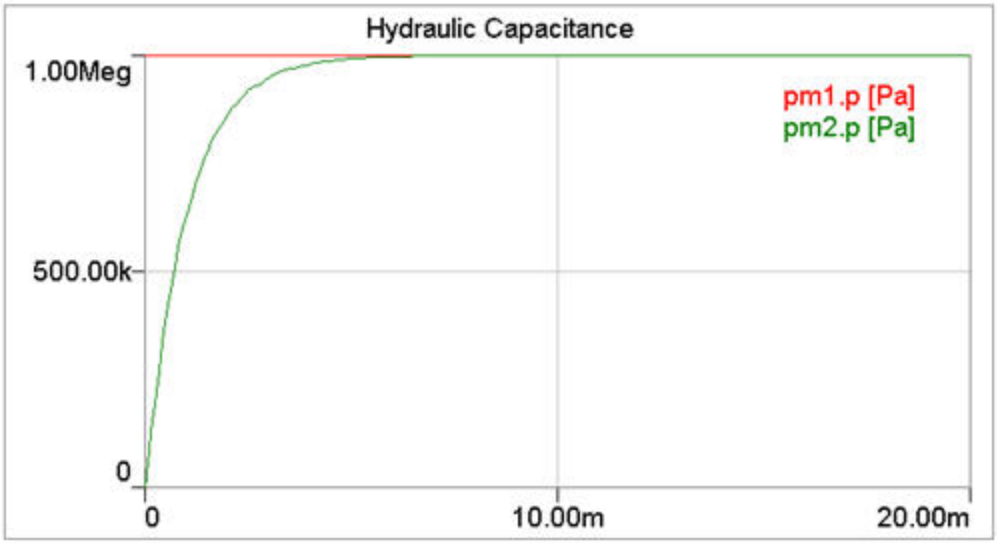


Figure 3. Simulation results-output showing volume being charged over time.

Top

References

[1] Hydraulic Control Systems, Herbert E. Merrit, John Wiley & Sons, Inc., Copyright 1967, Ref: Eqn. 3-53, page 52 (Basic compliance equation)

Hydraulic Inductance - VHDL-AMS

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents a hydraulic resistance that also includes the effects of fluid inertia, or hydraulic inductance. The characteristics are based on equations for laminar and turbulent flow through a smooth round pipe supplemented with the differential equation for effect of fluid mass and acceleration.

The Diameter, Length, and Fluid Density, dia, len, and rho respectively, can be a numerical values or defined by a pins.

Top

Assumptions and Limitations

Top

Mathematical Description

The Hydraulic Inductance model can be described as:

$$p(t) = L_{HYD} \cdot \frac{dq(t)}{dt}$$

$$L_{HYD} = \frac{rho \cdot len}{A}$$

$$A = \frac{\pi \cdot dia^2}{4}$$

where L_{HYD} is the equivalent hydraulic inductance.

Top

Netlist Syntax

COUPL lhyd ?InstanceName(@InstanceName):(@(Refbase)@(ID)) h1 := %0 , h2 := %1 (dia := @dia , len := @len , rho := @rho , q0 := @q0) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
h1	Plus Terminal	fluidic
h2	Minus Terminal	fluidic

Top

Parameters

Table 2

Name	Description	Data Type	Default Value[Unit]
rho	Density	real	900.0 [kg/m³]
dia	Diameter	real	1.0e-2 [m]
len	Length	real	1.0 [m]
q0	Initial Flow Rate	real	0.0 [m³/s]

Top

Example

This example illustrates the use of the Hydraulic Inductor model. The pressure source is increasing with time while the flow rate lags due to the inductor effects.

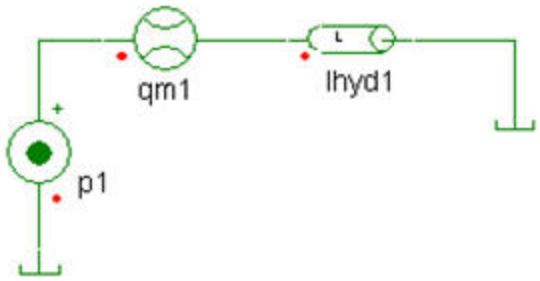


Figure 2. Application examples of the VHDL-AMS Hydraulic Inductor.

Table 3. System Parameters

Component	Parameter	Value [unit]
Fluidic Pressure Source p1	value	(100k)*t
Hydraulic Capacitance lhyd1	dia	1.0e-2 [m]
	len	1.0 [m]
	rho	900 [kg/m^3]
	q0	0 [m^3/s]

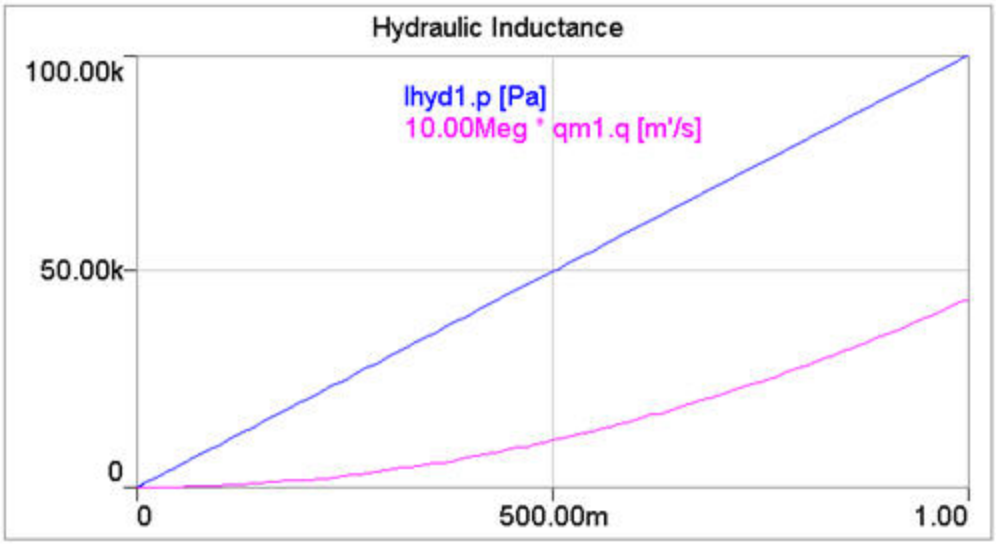


Figure 3. Simulation results-output showing increasing pressure from source and flow rate resulting from hydraulic inductor.

Top

References

Pressure Source - VHDL-AMS

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

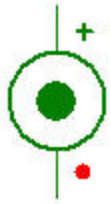


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents an ideal pressure source which supplies the defined pressure differential for an arbitrary flow rate.

Top

Assumptions and Limitations

The flow through an ideal pressure source can be arbitrary. Real pressure sources have always an internal resistance. If the internal resistance is essential for the simulation target, a corresponding resistance must be connected in series.

Top

Mathematical Description

Top

Netlist Syntax

COUPL p ?InstanceName(@InstanceName):(@(Refbase)@(ID)) h1 := %0 , h2 := %1 (value := @value , ac_mag := @ac_mag , ac_phase := @ac_phase) DST: SIM(Type:=SimVHDL) SRC:

DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature Type
h1	Plus Terminal	fluidic
h2	Minus Terminal	fluidic

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
value	Pressure	real	1.0e6 [Pa]
ac_mag	Magnitude of Pressure (AC)	real	1.0e-3 [Pa]
ac_ phase	Phase Shift of Pressure (AC)	real	0.0

Top

Example

This example illustrates the use of the Fluidic Pressure Source. The ideal source provides load independent pressure/flow rate. The time functions can be used as inputs for the source values.

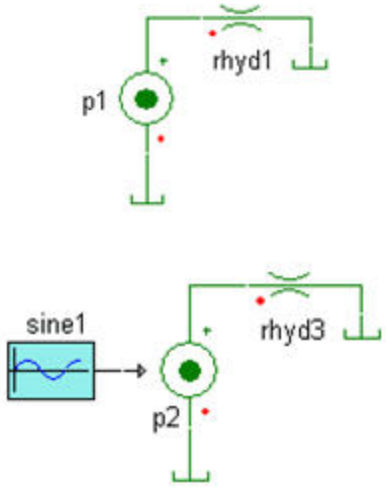


Figure 2. Application examples of the VHDL-AMS Fluidic Pressure Source

Table 3. System Parameters

Component	Parameter	Value [unit]
Sine Wave Time Function Block sine1	off	2Meg
	tdelay	0 [S]
	freq	5 [Hz]
	ampl	1Meg
Fluidic Pressure Source p1	value	1.0e+6
Fluidic Pressure Source p2	value	sine1.val
Linear Orifice rhyd1	k	1n
Linear Orifice rhyd3	k	5n

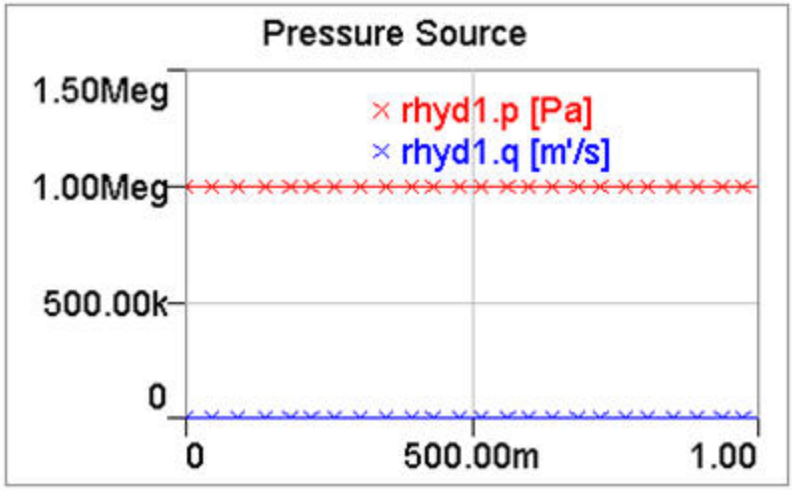


Figure 3. Simulation results-output showing pressure and flow generated by the pressure source (p1) in the orifice (rhyd1).

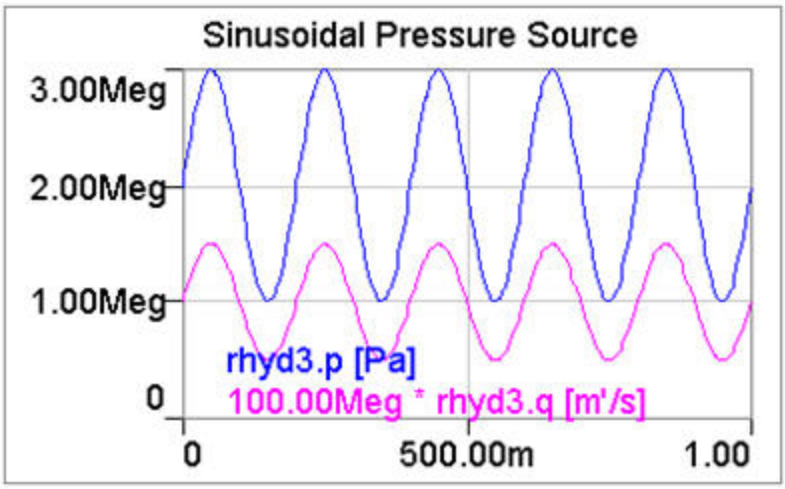


Figure 4. Simulation results-output showing pressure and flow generated by the pressure source (p2) in the orifice (rhyd3).

Top

References

Flow Source - VHDL-AMS

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents an ideal flow source which supplies the defined flow rate for an arbitrary pressure differential.

Top

Assumptions and Limitations

The pressure of an ideal flow source can be arbitrary. Real flow sources have always an internal resistance. If the internal resistance is essential for the simulation target, a corresponding resistance must be connected in parallel.

Top

Mathematical Description

Top

Netlist Syntax

COUPL q ?InstanceName(@InstanceName):(@(Refbase)@(ID)) h1 := %0 , h2 := %1 (value := @value , ac_mag := @ac_mag , ac_phase := @ac_phase) DST: SIM(Type:=SimVHDL) SRC:

DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
h1	Pin 1 (with red point)	fluidic
h2	Pin 2	fluidic

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
value	Flow Rate	real	1.0e-4 [m^3/s^2]
ac_mag	Magnitude of Flow Rate (AC)	real	1.0e-3
ac_ phase	Phase Shift of Flow Rate (AC)	real	0.0

Top

Example

This example illustrates the use of the Fluidic Flow Source. The ideal source provides load independent pressure/flow rate. The time functions can be used as inputs for the source values.

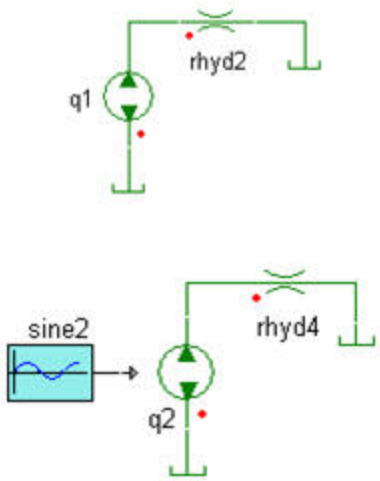


Figure 2. Application examples of the VHDL-AMS Fluidic Flow Source

Table 3. System Parameters

Component	Parameter	Value [unit]
Sine Wave Time Function Block sine2	off	0
	tdelay	0 [S]
	freq	5 [Hz]
	ampl	100u
Fluidic Flow Source q1	value	1.0e-4
Fluidic Flow Source q2	value	sine2.val
Linear Orifice rhyd2	k	5n
Linear Orifice rhyd4	k	5n

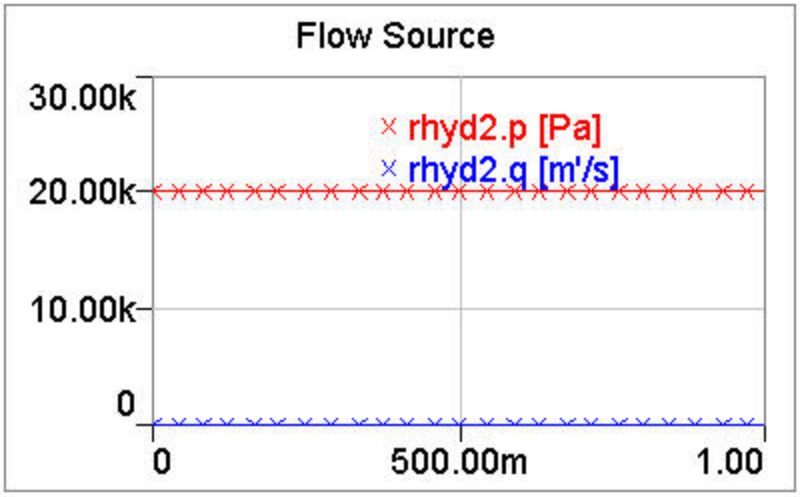


Figure 3. Simulation results-output showing pressure and flow generated by the flow source (q1) in the orifice (rhyd2).

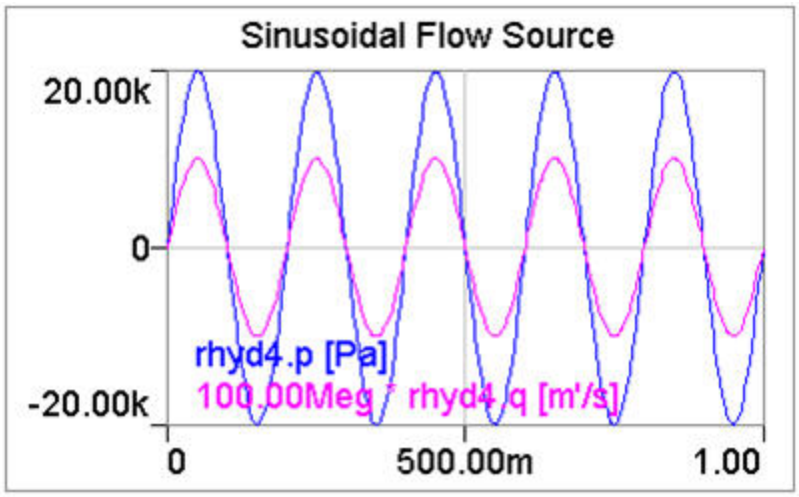


Figure 4. Simulation results-output showing pressure and flow generated by the flow source (q2) in the orifice (rhyd4).

Top

References

Hydraulic Resistance - VHDL-AMS

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

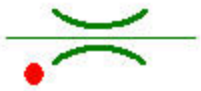


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents a simple fluid resistance specified by a linear coefficient.

Top

Assumptions and Limitations

Top

Mathematical Description

The Linear Orifice can be described as:

$$q(t) = k \cdot p(t)$$

where k is the Hydraulic Conductance.

Top

Netlist Syntax

COUPL rhyd ?InstanceName(@InstanceName):(@(Refbase)@(ID)) h1 := %0 , h2 := %1 (k := @k) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:-:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
h1	Plus Terminal	fluidic
h2	Minus Terminal	fluidic

Top

Parameters

Table 2

Name	Description	Data Type	Default Value[Unit]
k	Hydraulic Conductance	real	1.0e-9

Top

Example

This example illustrates the use of the Hydraulic Resistance with a Fluidic Pressure Source.

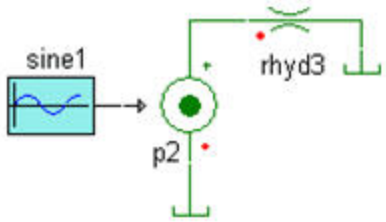


Figure 2. Application examples of the VHDL-AMS Hydraulic Resistance

Table 3. System Parameters

Component	Parameter	Value [unit]
Sine Wave Time Function Block sine1	off	2Meg
	tdelay	0 [S]
	freq	5 [Hz]
	ampl	1Meg
Fluidic Pressure Source p2	value	sine1.val
Linear Orifice rhyd3	k	5n

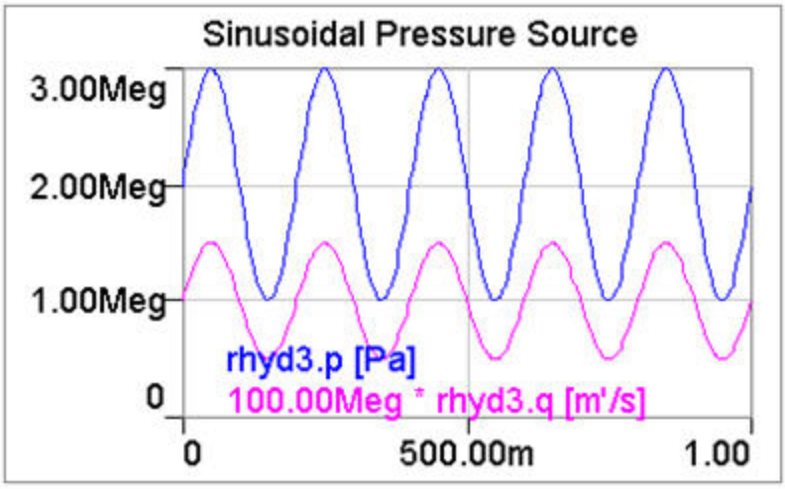


Figure 3. Simulation results-output showing pressure and flow generated by the pressure source (p2) in the orifice (rhyd3).

Top

References

Magnetic Components - VHDL-AMS

• Flux Source in VHDL-AMS (fluxsrc)

• Magneto-Motive Force Source in VHDL-AMS (mmfsrc)

• Magnetoreluctance in VHDL-AMS (rmag)

• Winding in VHDL-AMS (winding)

Across and Through Quantity of the Magnetic Domain

The magnetic domain components appear in the *Magnetic* folder of *Physical Domains* in the **Basic Elements VHDLAMS library** tab. The folder provides magnetic basic components to model simple magnetic equivalent circuits. The magnetic domain uses the magneto-motive force as across and the magnetic flux as through quantity.

By default, components and wires of the magnetic domain are represented in orange in the graphical representation of the Schematic. To change the nature colors, choose **Tools>Options>Schematic Editor Options**. Click on the Colors tab and select Wires as the object type, and the color of wires for different domains can be edited.

Across	Through	Equation
Magneto-Motive Force [A]	Magnetic Flux [Vs]	mv(t) = K * flux(t)

Flux Source - VHDL-AMS

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents an ideal magnetic flux source which supplies the defined magnetic flux for an arbitrary magneto-motive force.

Top

Assumptions and Limitations

The magneto-motive force of an ideal flux source can be arbitrary. Real flux sources have always an internal resistance. If the internal resistance is essential for the simulation target, a corresponding resistance must be connected in parallel.

Top

Mathematical Description

Top

Netlist Syntax

COUPL fluxsrc ?InstanceName(@InstanceName):(@(Refbase)@(ID)) mag1 := %0 , mag2 := %1 (value := @value , ac_mag := @ac_mag , ac_phase := @ac_phase) DST: SIM

(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:-
:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
mag1	Plus Terminal	magnetic
mag2	Minus Terminal	magnetic

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
value	Magnetic Flux	real	4.5e-3 [Vs]
ac_mag	Magnitude of flux (AC)	real	1.0e-3
ac_phase	Phase Shift of flux (AC)	real	0.0

Top

Example

This example illustrates the use of both the Magnetic Flux Source. The flux source is used with a Sine Wave Block to generate a sinusoidal magnetic flux which causes voltage induction in the winding. The magnetic power provided by the flux source is converted into heat by the resistor. Due to the ohmic losses at the winding, the power delivered to the resistor is lower than the power output of the flux source.

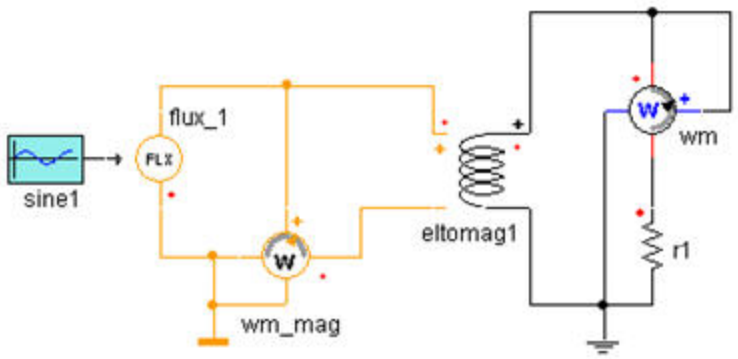


Figure 2. Application example of the VHDL-AMS Magnetic Flux Source.

Table 3. System Parameters

Component	Parameter	Value [unit]
Flux Source flux_1	value	sine1.val
Sine Wave Time Function Block sine1	ampl	4.5m
	tdelay	0 [S]
	freq	50 [Hz]
Winding eltomag1	flux0	0 [W]
	w	1000
	r	0.05 [Ohm]
Resistor r1	r	1 [Ohm]

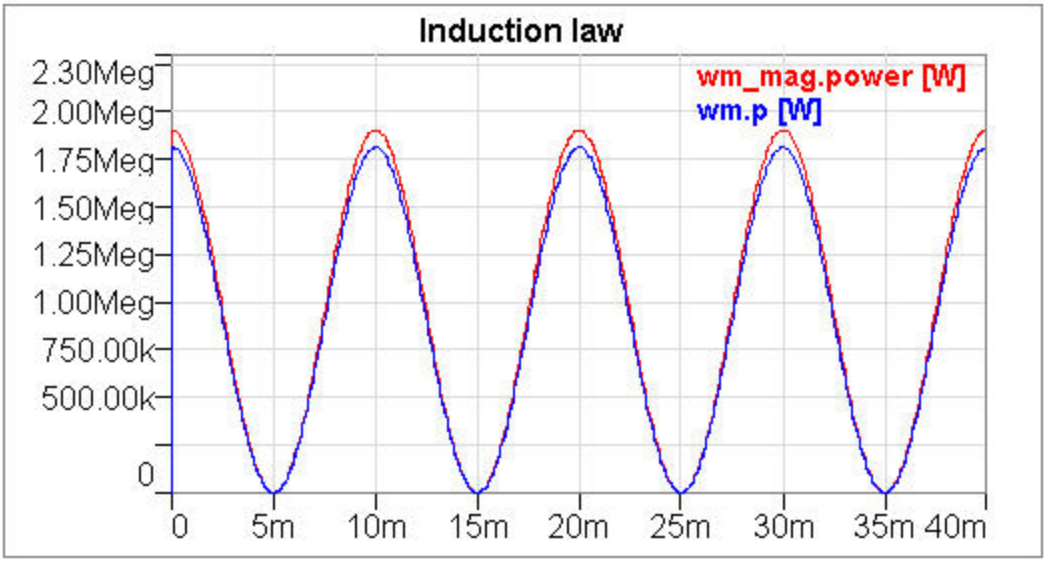


Figure 3. Simulation results-output showing power output of flux source and consumption of resistor(r1).

Top

References

Magneto-Motive Force Source - VHDL-AMS

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents an ideal magneto-motive force source which supplies the defined magnetic force for an arbitrary magnetic flux.

Top

Assumptions and Limitations

The magnetic flux of an ideal force source can be arbitrary. Real force sources have always an internal resistance. If the internal resistance is essential for the simulation target, a corresponding resistance must be connected in parallel.

Top

Mathematical Description

Top

Netlist Syntax

COUPL mmfsrc ?InstanceName(@InstanceName):(@(Refbase)@(ID)) mag1 := %0 , mag2 := %1 (value := @value , ac_mag := @ac_mag , ac_phase := @ac_phase) DST: SIM

(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:-
:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
mag1	Plus Terminal	magnetic
mag2	Minus Terminal	magnetic

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
value	Magnetomotive Force	real	1.0e4 [A]
ac_mag	Magnitude of MMF (AC)	real	1.0e-3 [A]
ac_ phase	Phase Shift of MMF (AC)	real	0.0

Top

Example

See Magnetic Circuit Example

References

Magnetoreluctance - VHDL-AMS

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents a magnetoreluctance that means a homogeneous distributed magnetic field within a media of proportional B-H-relation to model air gaps and leakage paths.

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

COUPL rmag ?InstanceName(@InstanceName):(@(Refbase)@(ID)) mag1 := %0 , mag2 := %1 (k := @k) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
mag1	Plus Terminal	magnetic
mag2	Minus Terminal	magnetic

Top

Parameters

Table 2

Name	Description	Data Type	Default Value[Unit]
k	Magnetic Resistance	real	1.0e9

Top

Example

See Magnetic Circuit Example

Top

References

Winding - VHDL-AMS

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

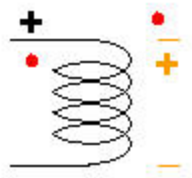


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents a coupling element between electric and magnetic circuit. It converts electrical energy into magnetic energy and vice versa (bidirectional energy flow). The component has electrical pins, n1 and n2 (across: v, through: i) and magnetic pins, mag1 and mag2. You can specify the number of turns (coupling quantity between magnetic and electric domain) and the ohmic winding resistance.

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

COUPL winding ?InstanceName(@InstanceName):(@(Refbase)@(ID)) n1 := %0 , n2 := %1 , mag1 := %2 , mag2 := %3 (flux0 := @flux0 , w := @w , r := @r) DST: SIM(Type:=SimVHDL)

SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
n1	Electrical Plus Terminal	electrical
n2	Electrical Minus Terminal	electrical
mag1	Magnetic Plus Terminal	magnetic
mag2	Magnetic Minus Terminal	magnetic

Top

Parameters

Table 2

Name	Description	Data Type	Default Value[Unit]
r	Resistance	resistance	1.0 [Ohm]
w	Number of Turns	real	1000.0
flux0	Initial Flux	flux	0.0

Top

Example

This example illustrates the use of both the Magnetic Winding model. A single phase transformer is modeled using the winding model together with a simple magnetic circuit. Although the turns ratio equals one, the secondary voltage is lower that the primary voltage due to copper losses.

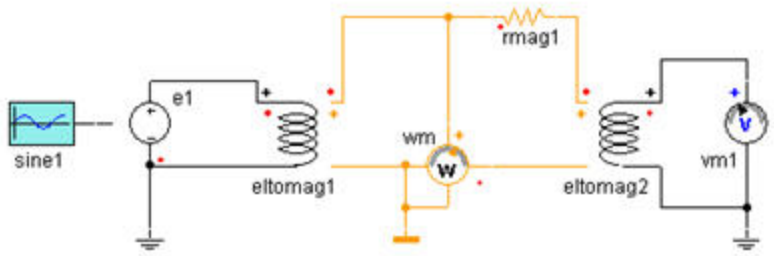


Figure 2. Application example of the VHDL-AMS Winding model.

Table 3. System Parameters

Component	Parameter	Value [unit]
Voltage Source e1	emf	sine1.val
Sine Wave Time Function Block sine1	ampl	326
	tdelay	0 [S]
	freq	50 [Hz]
Winding eltomag1/eltomag2	flux0	0 [W]
	w	1000
	r	3 [Ohm]
Magnetoreluctance rmag1	k	1.0e+8

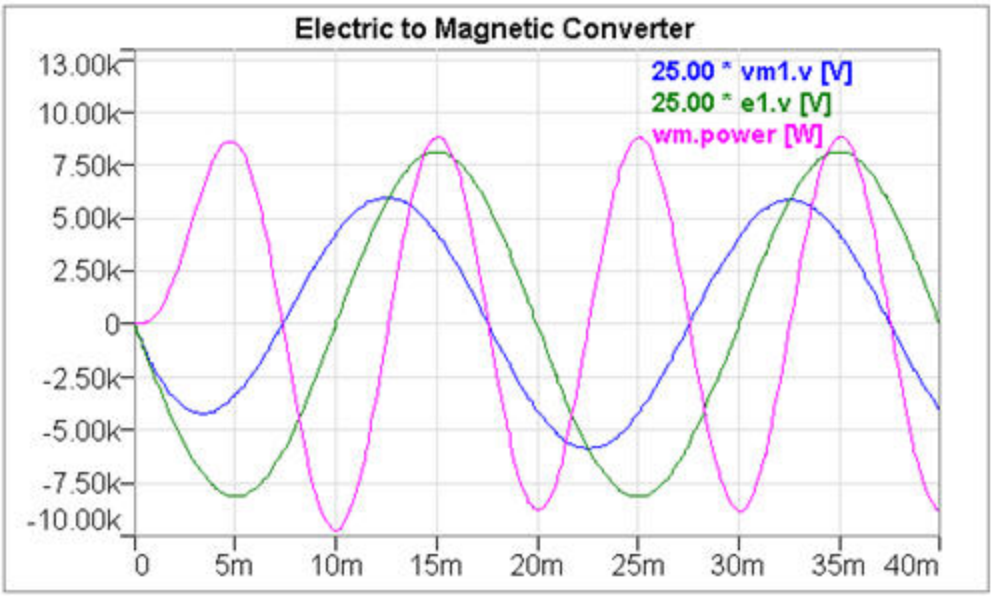


Figure 3. Simulation results-output showing input and output voltage and the power in the core.

Top

References

Magnetic Circuit Example - VHDL-AMS

This example illustrates the use of both the Magnetomotive Force source and the Magnetoresistor. The magnetic circuit of a three-phase system (e.g. transformer with open secondary) is modeled. Magnetic flux and mmf are measured at one particular point of the core.

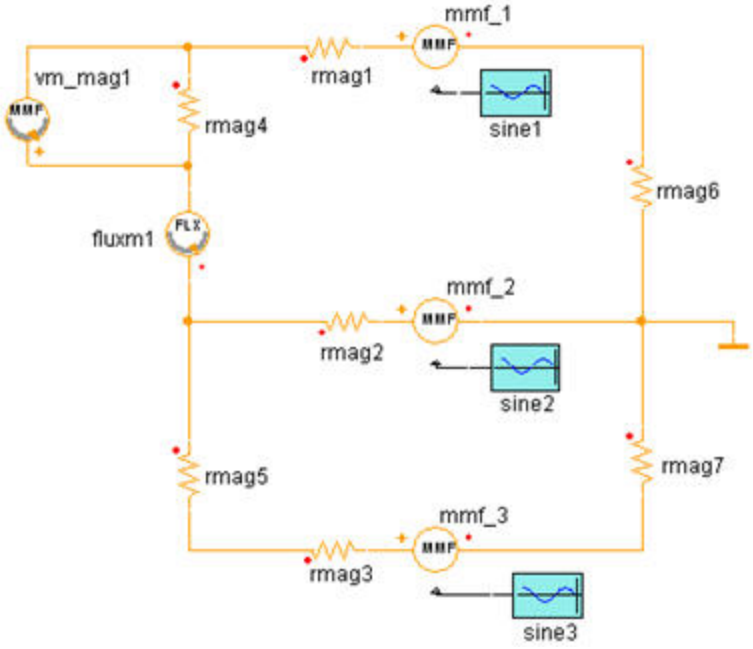


Figure 1. Application example of the VHDL-AMS Magnetomotive Force Source and Magnetoresistor.

Table 1. System Parameters

Component	Parameter	Value [unit]
Magnetomotive Force Source mmf_1/mmf_2/mmf_3	value	sine1.val/sine2.val/sine3.val
Sine Wave Time Function Block sine1/sine2/sine3	ampl	10k
	tdelay	0 [S]
	freq	50 [Hz]
	phase	0/(Pi*2/3)/(Pi*4/3)
Magnetoreluctance rmag1/rmag2/rmag3	k	1.0e+8
Magnetoreluctance rmag4/rmag5/rmag6/rmag7	k	5.0e+7

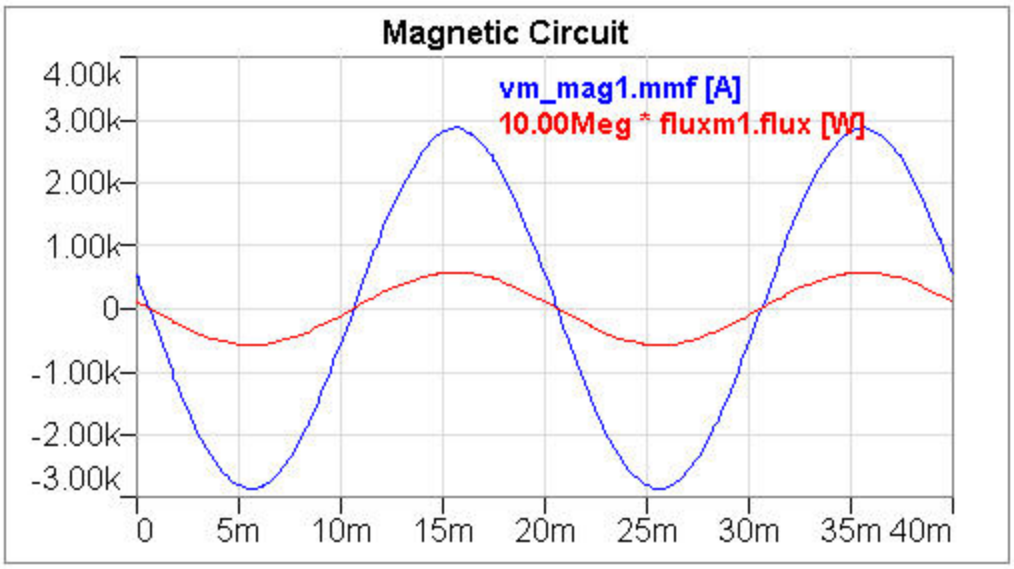


Figure 2. Simulation results-output showing mmf and flux generated by the circuit.

Top

References

Mechanical Components - VHDL-AMS

The mechanical domain components appear in the *Mechanical* folder of *Physical Domains* in the **Basic Element VHDLAMS** library. The folder provides mechanical basic components to model simple mechanical equivalent circuits. There are two representation forms:

- Displacement-Force
- Velocity-Force

The table shows the across and through quantities used of the representation forms.

By default, components and wires of the mechanical domain are represented in blue for translational systems and in lilac for rotational systems. To change the settings of nature colors, choose **Tools>Options>Schematic Editor Options**, click on the Colors tab, select Wires as the object type and the color of wires for different domains can be edited.

Across and Through Quantities of Mechanical Domains

Representation	System	Across	Through	Equation
Displacement-Force	Translational	s [m]	F [N]	v = s'dot
	Rotational	omega [rad]	M [Nm]	omega = phi'dot
Velocity-Force	Translational	v [m/s]	F [N]	s = v'integ
	Rotational	phi [rad/s]	M [Nm]	phi = omega'integ

Displacement-Force-Representation

The displacement-force representation consists of:

- Rotational Components
- Translational Components

Displacement-Force Rotational Components

- Damper in VHDL-AMS (damp_rotb)
- Torque Source in VHDL-AMS (f_rotb)
- Limit Stop in VHDL-AMS (limit_rotb)
- Mass in VHDL-AMS (mass_rotb)
- Angle Source in VHDL-AMS (s_rotb)
- Spacer in VHDL-AMS (spacer_rotb)
- Spring in VHDL-AMS (spring_rotb)
- Angular Velocity Source in VHDL-AMS (v_rotb)

Damper - VHDL-AMS

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

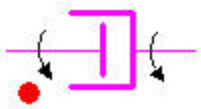


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents a simple damper specified by a linear damping coefficient.

Top

Assumptions and Limitations

Top

Mathematical Description

The Damper model can be described as:

$$m(t) = damping \cdot omega(t)$$

Where m(t) is the torque through terminal rotb1 to rotb2, and omega(t) is the angular velocity.

Top

Netlist Syntax

COUPL damp_rotb ?InstanceName(@InstanceName):(@(Refbase)@(ID)) rotb1 := %0 , rotb2 := %1 (damping := @damping) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
rotb1	Plus Terminal	rotational
rotb2	Minus Terminal	rotational

Top

Parameters

Table 2

Name	Description	Data Type	Default Value[Unit]
damping	Damping Coefficient	real	1.0 [Nm*s/rad]

Top

Example

See Angle Source/Spacer/Damper Rotational Displacement-Force Example

Top

References

Torque Source - VHDL-AMS

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Input/Output Quantities

• Example

• References

Description

The component represents an ideal mechanical force/torque source which supplies the defined force/torque for an arbitrary position.

Top

Assumptions and Limitations

The position of an ideal force/torque source can be arbitrary. Real flow sources have always an internal resistance. If the internal resistance is essential for the simulation target, a corresponding resistance must be connected in parallel.

Top

Mathematical Description

Top

Netlist Syntax

COUPL f_rotb ?InstanceName(@InstanceName):(@(Refbase)@(ID)) rotb1 := %0 , rotb2 := %1 (value := @value , ac_mag := @ac_mag , ac_phase := @ac_phase) DST: SIM(Type:-:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
rotb1	Plus Terminal	rotational
rotb2	Minus Terminal	rotational

Top

Parameters

Table 2

Name	Description	Data Type	Default Value[Unit]
value	Torque	real	1.0 [Nm]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
ac_mag	Magnitude of Torque (AC)		real
ac_phase	Phase Shift of Torque (AC)		real

Top

Example

See Torque Source/Mass/Limit Stop Rotational Displacement-Force Example

Top

References

Limit Stop - VHDL-AMS

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Input/Output Quantities

• Example

• References

Description

The component represents a parallel damper and spring combination within defined limits (upper and lower limit).

Top

Assumptions and Limitations

Top

Mathematical Description

The Limit Stop can be described as:

If phi(t) > phiul, m(t) = c × [phi(t) - phiul] + damping × omega(t)]

If phi(t) < phiul, m(t) = c × [phi(t) – phill] + damping × omega(t)]

Else m(t) = 0

where m(t) and phi(t) are the torque through and angle across terminal rotb1 and rotb2, respectively.

Top

Netlist Syntax

COUPL limit_rotb ?InstanceName(@InstanceName):(@(Refbase)@(ID)) rotb1 := %0 , rotb2 := %1 (c := @c , damping := @damping , phiul := @phiul , phill := @phill) DST: SIM(Type:-:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
rotb1	Plus Terminal	rotational
rotb2	Minus Terminal	rotational

Top

Parameters

Table 2

Name	Description	Data Type	Default Value[Unit]
c	Spring Rate	real	1.0e6 [Nm/rad]
damping	Damping Coefficient	real	1.0e6 [Nm*s/rad]
phiul	Upper Angle Limit	real	1.0 [rad]
phill	Lower Angle Limit	real	0.0 [rad]

Top

Input/Output Quantities

Top

Example

See Torque Source/Mass/Limit Stop Rotational Displacement-Force Example

Top

References

Mass - VHDL-AMS

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

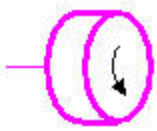


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents a mass specified by a mass/inertia and initial position/velocity.

Initial Position, and Initial Velocity are of the VHDL data type generic, therefore, their numerical values are defined at t = 0 and remain unchanged for the remainder of the simulation.

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

COUPL mass_rotb ?InstanceName(@InstanceName):(@(Refbase)@(ID)) rotb1 := %0 (phi0 := @phi0 , omega0 := @omega0 , j := @j) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
rotb1	Plus Terminal	rotational

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
j	Moment of Inertia	real	1.0 [kg*m^2]
phi0	Initial Angle	real	0.0 [rad]
omega0	Initial Angular Velocity	real	0.0 [rad/s]

Top

Example

See Torque Source/Mass/Limit Stop Rotational Displacement-Force Example

Top

References

Angle Source - VHDL-AMS

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

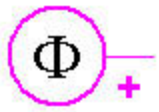


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents an ideal mechanical position source which supplies the defined position for an arbitrary force/torque.

Top

Assumptions and Limitations

The force/torque through an ideal position/angle source can be arbitrary. Real force/torque sources have always an internal resistance. If the internal resistance is essential for the simulation target, a corresponding resistance must be connected in series.

Top

Mathematical Description

Top

Netlist Syntax

COUPL s_rotb ?InstanceName(@InstanceName):(@(Refbase)@(ID)) rotb1 := %0 (value := @value , ac_mag := @ac_mag , ac_phase := @ac_phase) DST: SIM(Type:=SimVHDL) SRC:

DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
rotb1	Plus Terminal	rotational

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
value	Angle	real	1.0 [rad]
ac_mag	Magnitude of Angle (AC)	real	1.0e-3 [rad]
ac_ phase	Phase Shift of Angle (AC)	real	0.0 [rad]

Top

Example

See Angle Source/Spacer/Damper Rotational Displacement-Force Example

Top

References

Spacer - VHDL-AMS

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

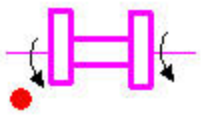


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents a constant displacement between two components in a mechanical system specified by a length.

Top

Assumptions and Limitations

Top

Mathematical Description

The Spacer model can be described as:

$$omega(t) = \frac{d[angle(t)]}{dt}$$

Where omega(t) is the angular velocity of the spacer.

Top

Netlist Syntax

COUPL spacer_rotb ?InstanceName(@InstanceName):(@(Refbase)@(ID)) rotb1 := %0 , rotb2 := %1 (angle := @angle) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
rotb1	Plus Terminal	rotational
rotb2	Minus Terminal	rotational

Top

Parameters

Table 2

Name	Description	Data Type	Default Value[Unit]
angle	Angular Displacement	real	0.0 [rad]

Top

Example

See Angle Source/Spacer/Damper Rotational Displacement-Force Example

Top

References

Spring - VHDL-AMS

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents a simple spring specified by a spring rate.

Top

Assumptions and Limitations

Top

Mathematical Description

The Spring model can be described as:

$$m(t) = c \cdot phi(t)$$

Where m(t) and phi(t) are the torque through and the angle across the terminal rotb1 and rotb2.

Top

Netlist Syntax

COUPL spring_rotb ?InstanceName(@InstanceName):(@(Refbase)@(ID)) rotb1 := %0 , rotb2 := %1 (c := @c) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:-:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
rotb1	Plus Terminal	rotational
rotb2	Minus Terminal	rotational

Top

Parameters

Table 2

Name	Description	Data Type	Default Value[Unit]
c	Spring Rate	real	1.0 [Nm/rad]
phi0	Initial Angle	real	0 [deg]

Top

Example

See Angular Velocity Source/Spring/Damper Rotational Displacement-Force Example

Top

References

Angular Velocity Source - VHDL-AMS

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

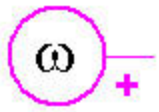


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents an ideal mechanical velocity source which supplies the defined velocity for an arbitrary force/torque.

Top

Assumptions and Limitations

The force/torque through an ideal position/angle source can be arbitrary. Real force/torque sources have always an internal resistance. If the internal resistance is essential for the simulation target, a corresponding resistance must be connected in series.

Top

Mathematical Description

Top

Netlist Syntax

COUPL v_rotb ?InstanceName(@InstanceName):(@(Refbase)@(ID)) rotb1 := %0 (phi0 := @phi0 , value := @value , ac_mag := @ac_mag , ac_phase := @ac_phase) DST: SIM

(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:-
:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
rotb1	Plus Terminal	rotational

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
value	Angular Velocity [rad/s]	real	1.0 [rad/s]
ac_mag	Magnitude of Angular Velocity (AC)	real	1.0e-3 [rad/s]
ac_ phase	Phase Shift of Angular Velocity (AC)	real	0.0 [rad]
phi0	Initial Angle	real	0.0 [rad]

Top

Example

See Angular Velocity Source/Spring/Damper Rotational Displacement-Force Example

Top

References

Angle Source/Spacer/Damper Rotational Displacement-Force Example

The first example is a displaced mass with spacer. It combines an angle source, spacer, spring, damper and mass model into a single example. In this example, a sinusoidal displacement is applied to the mass with the source offset removed with the spacer.

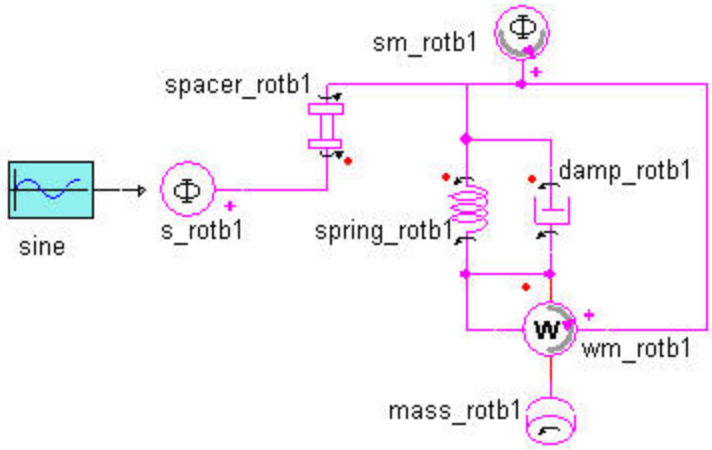


Figure 1. Application examples of the VHDL-AMS Rotational DF Angle Source with Spacer and Damper

Table 1. System Parameters

Component	Parameter	Value [unit]
Sine Wave Time Function Block sine1	off	0.2
	tdelay	0 [S]
	freq	1 [Hz]
	ampl	1.5
Angle Source(Rotational DF) s_rotb1	value	sine1.val
Spacer (Rotational DF) spacer_rotb1	angle	0.2
Spring (Rotational DF) spring_rotb1	c	10.0 [Nm/rad]

Damper (Rotational DF) damp_rotb1	damping	0.05 [Ns/m]
Mass (Rotational DF) mass_rotb1	phi0	0.0 [rad]
	omega0	0.0 [rad/s]
	j	1.0 [kg*m%]

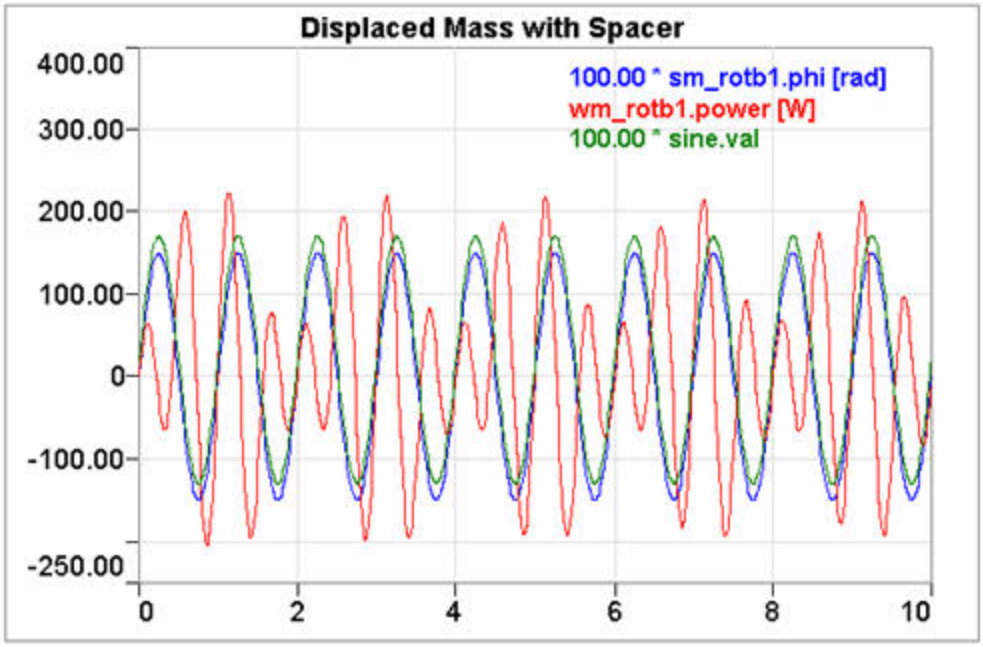


Figure 2. Simulation results-showing sine input, output of position source and power delivered to mass.

Angular Velocity Source/Spring/Damper Rotational Displacement-Force Example

This example is an underdamped spring/mass system. It combines an angular velocity source, spring, damper and mass model into a single example. In this example, a constant angular velocity is applied to the system. The damping causes the force applied to the mass element to decay sinusoidally over time.

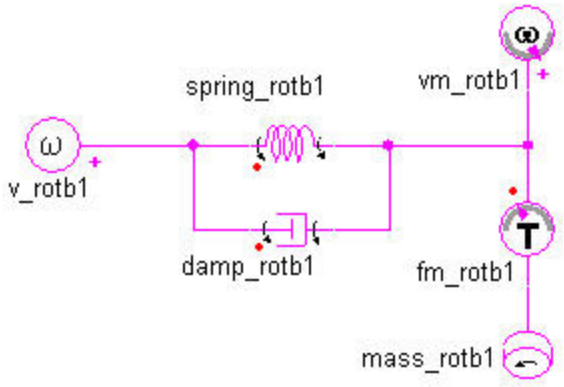


Figure 1. Application examples of the VHDL-AMS Rotational DF Angular Velocity Source with Spring and Damper

Table 1. System Parameters

Component	Parameter	Value [unit]
Angular Velocity Source(Rotational DF) v_rotb1	value	10 [rad/s]
Spring (Rotational DF) spring_rotb1	c	10 [Nm/rad]
Damper (Rotational DF) damp_rotb1	damping	0.05 [Ns/m]
Mass (Rotational DF) mass_rotb1	phi0	0.0 [rad]
	omega0	0.0 [rad/s]
	j	1.0

		[kg*m%]

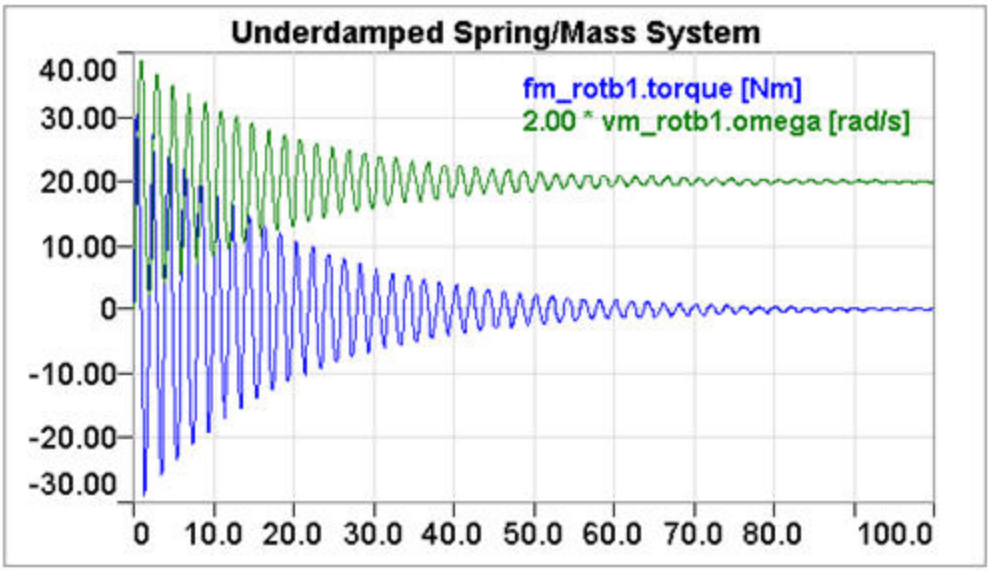


Figure 2. Simulation results-showing velocity and force delivered to mass decaying with time.

Torque Source/Mass/Limit Stop Rotational Displacement-Force Example

This example is a mass accelerated by a constant torque. The acceleration is limited due to the Limit Stop. The Limit Stop represents a spring-damper combination which causes the mass to stop at the defined upper angle limit.

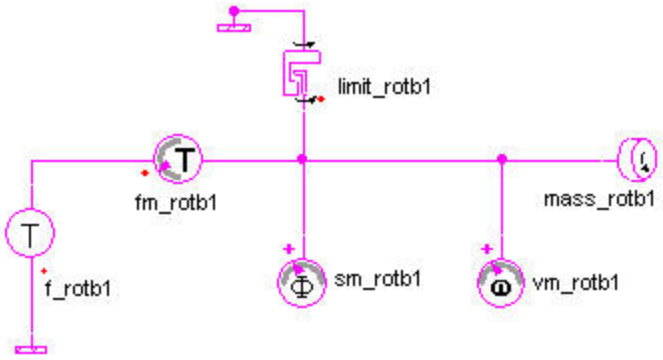


Figure 1. Application examples of the VHDL-AMS Rotational DF Torque Source with Mass and Limit Stop

Table 1. System Parameters

Component	Parameter	Value [unit]
Torque Source(Rotational DF) f_rotb1	value	10 [Nm]
Limit Stop (Rotational DF) limit_rotb1	c	1e6 [N/m]
	phiul	1.0 [rad]
	damping	1e6 [Ns/m]
	phill	0 [rad]
Mass (Rotational DF) mass_rotb1	phi0	0.0 [rad]
	omega0	0.0 [rad/s]
	j	3 [kg*m%]

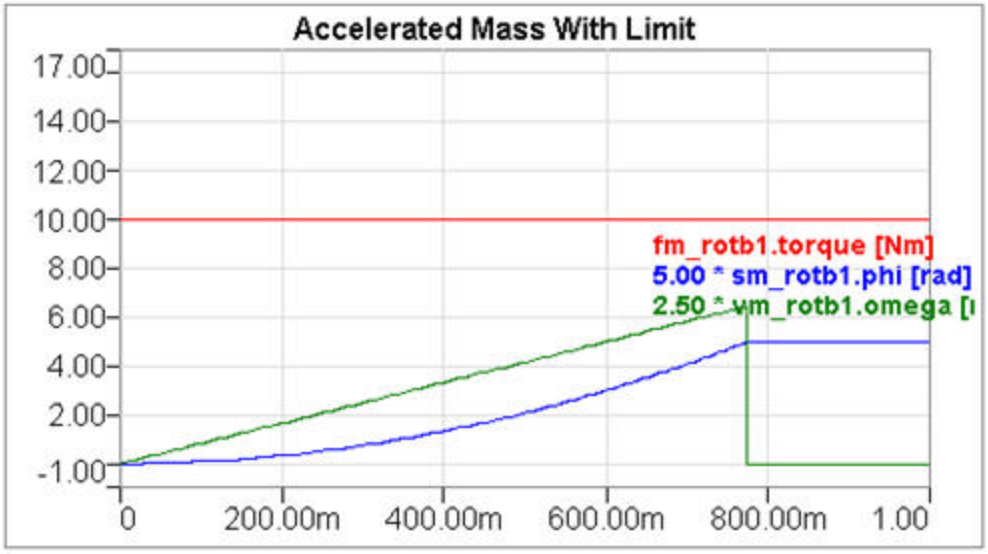


Figure 2. Simulation results-showing velocity and position change up to the limit, as well as, the constant input force.

Displacement-Force Translational Components

- Damper in VHDL-AMS (damp_trb)
- Force Source in VHDL-AMS (f_trb)
- Limit Stop in VHDL-AMS (limit_trb)
- Mass in VHDL-AMS (mass_trb)
- Position Source in VHDL-AMS (s_trb)
- Solenoid Model in VHDL-AMS (solenoid_trb)
- Spacer in VHDL-AMS (spacer_trb)
- Spring in VHDL-AMS (spring_trb)
- Velocity Source in VHDL-AMS (v_trb)

Damper - VHDL-AMS

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

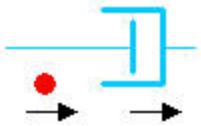


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents a simple damper specified by a linear damping coefficient. The Damping Coefficient, damping, can be a numerical value or defined by a pin.

Top

Assumptions and Limitations

Top

Mathematical Description

The Damper model can be described as:

$$F(t) = damping \cdot v(t)$$

Where F(t) is the force through terminal trb1 to mechanical reference, and v(t) is the velocity.

Top

Netlist Syntax

COUPL damp_trb ?InstanceName(@InstanceName):(@(Refbase)@(ID)) trb1 := %0 , trb2 := %1 (damping := @damping) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
trb1	Plus Terminal	translational
trb2	Minus Terminal	translational

Top

Parameters

Table 2

Name	Description	Data Type	Default Value[Unit]
damping	Damping Coefficient	real	1.0 [Ns/m]

Top

Example

See Velocity Source/Spring/Damper Translational Displacement-Force Example

Top

References

Force Source - VHDL-AMS

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Input/Output Quantities

• Example

• References

Description

The component represents an ideal mechanical force/torque source which supplies the defined force/torque for an arbitrary position.

Top

Assumptions and Limitations

The position of an ideal force/torque source can be arbitrary. Real flow sources have always an internal resistance. If the internal resistance is essential for the simulation target, a corresponding resistance must be connected in parallel.

Top

Mathematical Description

Top

Netlist Syntax

COUPL f_trb ?InstanceName(@InstanceName):(@(Refbase)@(ID)) trb1 := %0 , trb2 := %1 (value := @value , ac_mag := @ac_mag , ac_phase := @ac_phase) DST: SIM(Type:-:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
trb1	Plus Terminal	translational
trb2	Minus Terminal	translational

Top

Parameters

Table 2

Name	Description	Data Type	Default Value[Unit]
value	Force	real	1.0 [N]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
s/phi	Displacement [m]	Output	real
ac_mag	Magnitude of Force	Output	real
ac_phase	Phase Shift of Force	Output	real

Top

Example

See Force Source/Mass/Limit Stop Translational Displacement-Force Example

Top

References

Limit Stop - VHDL-AMS

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

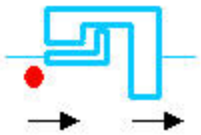


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Input/Output Quantities

• Example

• References

Description

The component represents a parallel damper and spring combination within defined limits (upper and lower limit).

Top

Assumptions and Limitations

Top

Mathematical Description

The Limit Stop can be described as:

If s(t) > sul, F(t) = c × [s(t) - sul] + damping × v(t)

If s(t) < sul, F(t) = c × [s(t) – sll_val] + damping × v(t)

Else F(t) = 0

where F(t) and s(t) are the force through and displacement across terminal trb1 and trb2.

Top

Netlist Syntax

COUPL limit_trb ?InstanceName(@InstanceName):(@(Refbase)@(ID)) trb1 := %0 , trb2 := %1 (c := @c , damping := @damping , sul := @sul , sll_val := @sll_val) DST: SIM(Type:-:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
trb1	Plus Terminal	translational
trb2	Minus Terminal	translational

Top

Parameters

Table 2

Name	Description	Data Type	Default Value[Unit]
c	Spring Rate	real	1.0e6 [N/m]
damping	Damping Coefficient	real	1.0e6 [Ns/m]
sul	Upper Position Limit	real	1.0 [m]
sll_val	Lower Position Limit	real	0.0 [m]

Top

Input/Output Quantities

Top

Example

See Force Source/Mass/Limit Stop Translational Displacement-Force Example

Top

References

Mass - VHDL-AMS

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents a mass specified by a mass/inertia and initial position/velocity.

Initial Position, and Initial Velocity are of the VHDL data type generic, therefore, their numerical values are defined at t = 0 and remain unchanged for the remainder of the simulation.

Top

Assumptions and Limitations

Top

Mathematical Description

The Mass model can be described as:

$$F(t) = M \cdot a(t)$$

Where F(t) is the force through terminal trb1 to mechanical reference, and a(t) is the acceleration of the mass.

Top

Netlist Syntax

COUPL mass_trb ?InstanceName(@InstanceName):(@(Refbase)@(ID)) trb1 := %0 (s0 := @s0 , v0 := @v0 , m := @m) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
trb1	Plus Terminal	translational

Top

Parameters

Table 2

Name	Description	Data Type	Default Value[Unit]
m	Mass	real	1.0 [kg]
s0	Initial Displacement	real	0.0 [m]
v0	Initial Velocity	real	0.0 [m/s]

Top

Example

See Force Source/Mass/Limit Stop Translational Displacement-Force Example

Top

References

Position Source - VHDL-AMS

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents an ideal mechanical position source which supplies the defined position for an arbitrary force/torque.

Top

Assumptions and Limitations

The force/torque through an ideal position/angle source can be arbitrary. Real force/torque sources have always an internal resistance. If the internal resistance is essential for the simulation target, a corresponding resistance must be connected in series.

Top

Mathematical Description

Top

Netlist Syntax

COUPL s_trb ?InstanceName(@InstanceName):(@(Refbase)@(ID)) trb1 := %0 (value := @value , ac_mag := @ac_mag , ac_phase := @ac_phase) DST: SIM(Type:=SimVHDL) SRC:

DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
trb1	Plus Terminal	translational

Top

Parameters

Table 2

Name	Description	Data Type	Default Value[Unit]
value	Displacement	real	1.0 [m]
ac_mag	Magnitude of Position	real	1.0e-3 [m]
ac_phase	Phase Shift of Position	real	0.0

Top

Example

See Position/Spacer/Damper Translational Displacement-Force Example

Top

References

AMS Solenoid

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

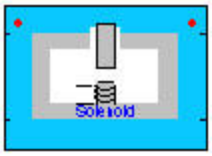


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The Solenoid component is an ideal behavioral model of a linear actuator.

Top

Assumptions and Limitations

Top

Mathematical Description

The inductance and generated force of the solenoid are described by the following equations:

$$L(x) = n^2 \cdot \left(\frac{l0}{1 + k \cdot x} \right)$$

$$F = \frac{-(n^2 \cdot L0 \cdot k \cdot i^2)}{2 \cdot (1 + k \cdot x)^2}$$

where x is the plunger gap, l0 is the maximum inductance per turn, k is the inductance coefficient, n is the number of coil turns, and I is the solenoid current.

Top

Netlist Syntax

COUPL solenoid_trb ?InstanceName(@InstanceName):(@(Refbase)@(ID)) p := %0 , m := %1 , pos1 := %2 , pos2 := %3 (l0 := @l0 , k := @k , n := @n) DST: SIM(Type:=SimVHDL) SRC: DB (File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
p	Positive electrical terminal	electrical
m	Negative electrical terminal	electrical
pos1	Postive mechanical termainal	translational
pos2	Negative mechanical terminal	translational

Top

Parameters

Table 2

Name	Description	Data Type	Default Value[Unit]
l0	Maximum inductance per turn at the minimum gap	real	1.25e-7 [H]
k	Inductance coefficient	real	197.735 [1/m]
n	Number of coil turns	real	1.0

Top

Example

Top

References

Spacer - VHDL-AMS

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

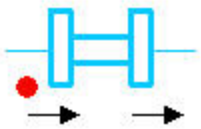


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents a constant displacement between two components in a mechanical system specified by a length.

Top

Assumptions and Limitations

Top

Mathematical Description

The Spacer model can be described as:

$$v(t) = \frac{d(len)}{dt}$$

Where v(t) is the velocity of the spacer.

Top

Netlist Syntax

COUPL spacer_trb ?InstanceName(@InstanceName):(@(Refbase)@(ID)) trb1 := %0 , trb2 := %1 (len := @len) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:-:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
trb1	Plus Terminal	translational
trb2	Minus Terminal	translational

Top

Parameters

Table 2

Name	Description	Data Type	Default Value[Unit]
len	Length	real	0.0

Top

Example

See Position Source/Spacer/Damper Translational Displacement-Force Example

Top

References

Spring - VHDL-AMS

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

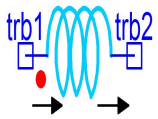


Figure 1. Component symbol

• Description
• Assumptions and Limitations
• Mathematical Description
• Netlist Syntax
• Conservative Pins
• Parameters
• Input/Output Quantities
• Example
• References

Description

The component represents a simple spring specified by a spring rate.

Top

Assumptions and Limitations

Top

Mathematical Description

The Spring model can be described as:

$$f(t) = c \cdot s(t)$$

Where f(t) and s(t) are the force through and the displacement across the terminal trb1 and trb2.

Top

Netlist Syntax

COUPL spring_trb ?InstanceName(@InstanceName):(@(Refbase)@(ID)) trb1 := %0 , trb2 := %1 (c := @c) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:-:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
trb1	Plus Terminal	translational
trb2	Minus Terminal	translational

Top

Parameters

Table 2

Name	Description	Data Type	Default Value[Unit]
c	Spring Rate	real	100.0 [N/m]
s0	Initial Displacement	real	0.0 [m]

Top

Input/Output Quantities

Top

Example

See Velocity Source/Spring/Damper Translational Displacement-Force Example

Top

References

Velocity Source - VHDL-AMS

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

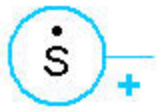


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents an ideal mechanical velocity source which supplies the defined velocity for an arbitrary force/torque.

Top

Assumptions and Limitations

The force/torque through an ideal position/angle source can be arbitrary. Real force/torque sources have always an internal resistance. If the internal resistance is essential for the simulation target, a corresponding resistance must be connected in series.

Top

Mathematical Description

Top

Netlist Syntax

COUPL v_trb ?InstanceName(@InstanceName):(@(Refbase)@(ID)) trb1 := %0 (s0 := @s0 , value := @value , ac_mag := @ac_mag , ac_phase := @ac_phase) DST: SIM

(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:-
:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
trb1	Plus Terminal	translational

Top

Parameters

Table 2

Name	Description	Data Type	Default Value[Unit]
value	Velocity	real	1.0 [m/s]
ac_mag	Magnitude of Velocity	real	1.0e-3 [m]
ac_phase	Phase Shift of Velocity	real	0.0
s0	Initial Position	real	0.0 [m]

Top

Example

See Velocity Source/Spring/Damper Translation Displacement-Force Example

Top

References

Position Source/Spacer/Damper Translational Displacement-Force Example

The first example is a displaced mass with spacer. It combines a position source, spacer, damper and mass model into a single example. In this example, a sinusoidal displacement is applied to the mass with the source offset removed with the spacer.

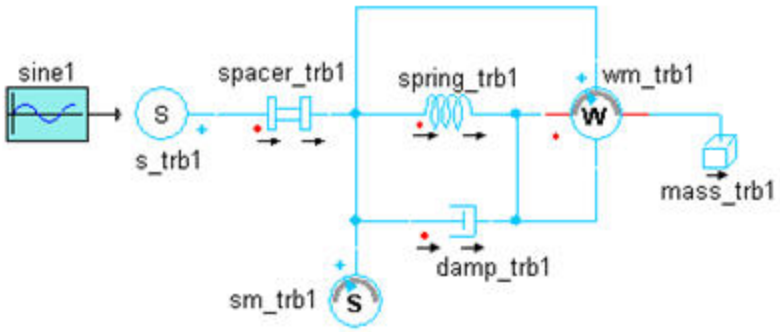


Figure 1. Application examples of the VHDL-AMS Translational DF Position Source with Spacer and Damper

Table 1. System Parameters

Component	Parameter	Value [unit]
Sine Wave Time Function Block sine1	off	1
	tdelay	0 [S]
	freq	1 [Hz]
	ampl	1
Postion Source(Translational DF) s_ trb1	value	sine1.val
Spacer (Translational DF) spacer_trb1	len	1
Spring (Translational DF) spring_trb1	c	10
Damper (Translational DF) damp_trb1	damping	1.0
Mass (Translational DF) mass_trb1	s0	0.0 [m]
	v0	0.0 [m/s]
	m	1.0 [kg]

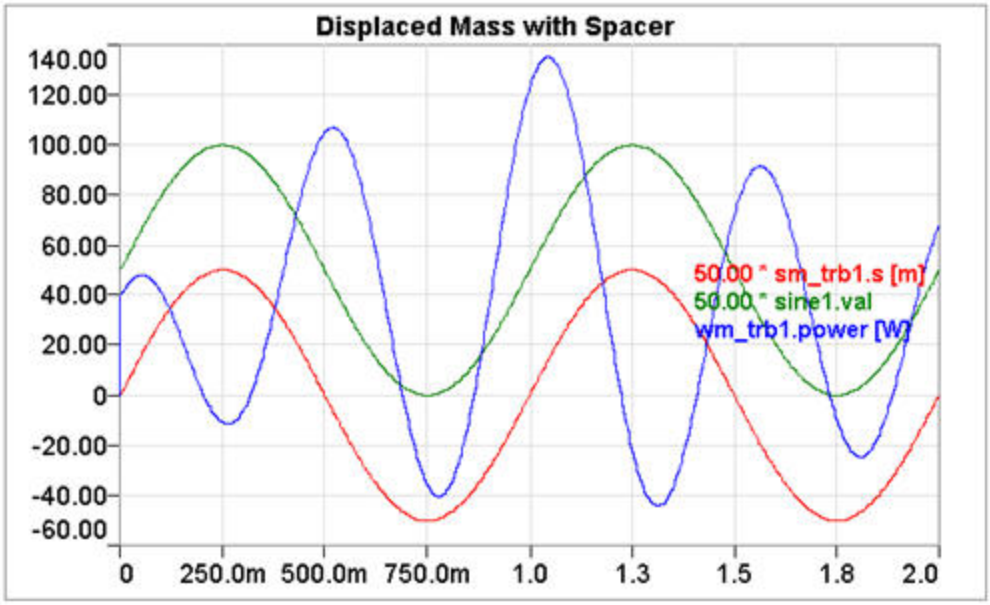


Figure 2. Simulation results-showing output of position source, spacer output, and power delivered to mass.

Velocity Source/Spring/Damper Translational Displacement-Force Example

This example is an underdamped spring/mass system. It combines a velocity source, spring, damper and mass model into a single example. In this example, a constant velocity is applied to the system. The damping causes the force applied to the mass element to decay sinusoidally over time.

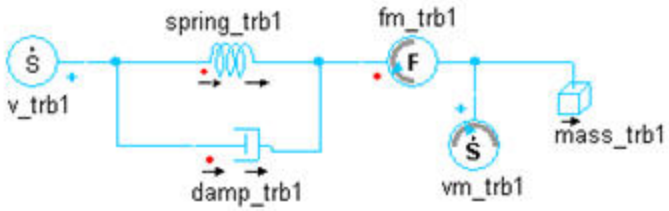


Figure 1. Application examples of the VHDL-AMS Translational DF Velocity Source with Spring and Damper

Table 1. System Parameters

Component	Parameter	Value [unit]
Velocity Source(Translational DF) v_ trb1	value	10 [m/s]
Spring (Translational DF) spring_trb1	c	10k
Damper (Translational DF) damp_trb1	damping	5.0
Mass (Translational DF) mass_trb1	s0	0.0 [m]
	v0	0.0 [m/s]
	m	500.0 [kg]

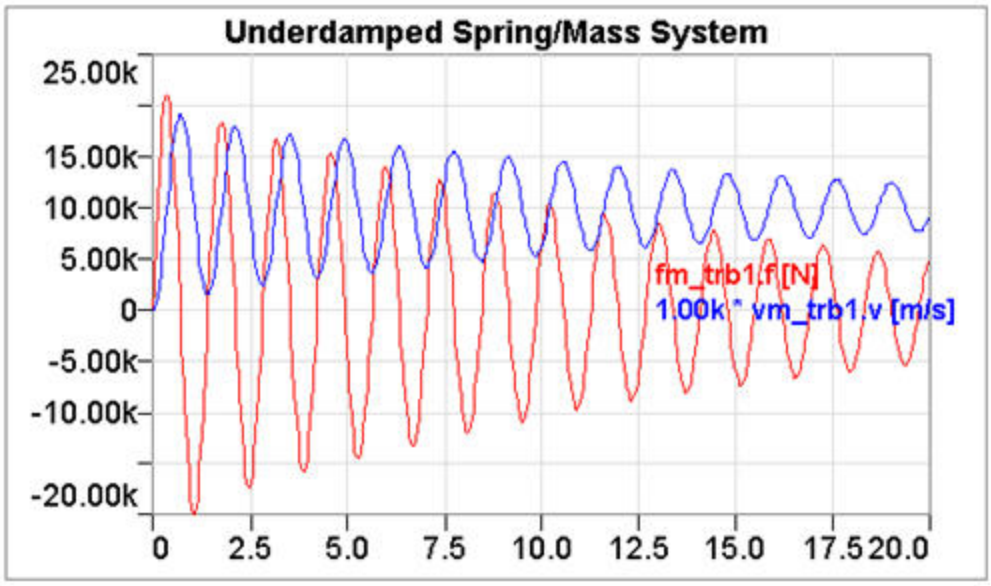


Figure 2. Simulation results-showing velocity and force delivered to mass.

Force Source/Mass/Limit Stop Translational Displacement-Force Example

This example is a mass accelerated by a constant force. The acceleration is limited due to the Limit Stop. The Limit Stop represents a spring-damper combination which causes the mass to stop at the defined upper position.

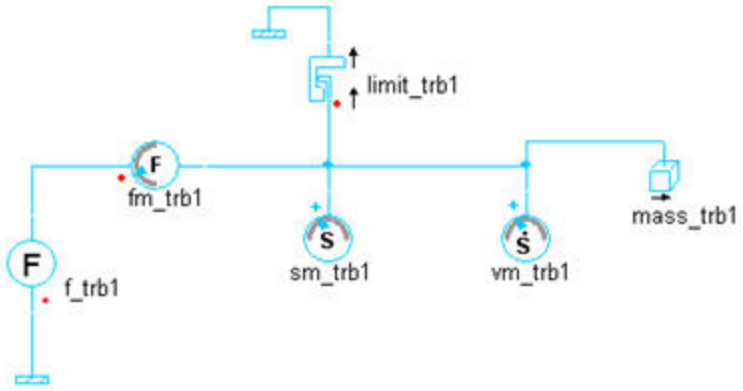


Figure 1. Application examples of the VHDL-AMS Translational DF Force Source with Mass and Limit Stop

Table 1. System Parameters

Component	Parameter	Value [unit]
Force Source(Translational DF) f_trb1	value	10 [N]
Limit Stop (Translational DF) limit_trb1	c	1e5 [N/m]
	sll_val	0
	damping	1e5 [Ns/m]
	sul	1
Mass (Translational DF) mass_trb1	s0	0.0 [m]
	v0	0.0 [m/s]
	m	3 [kg]

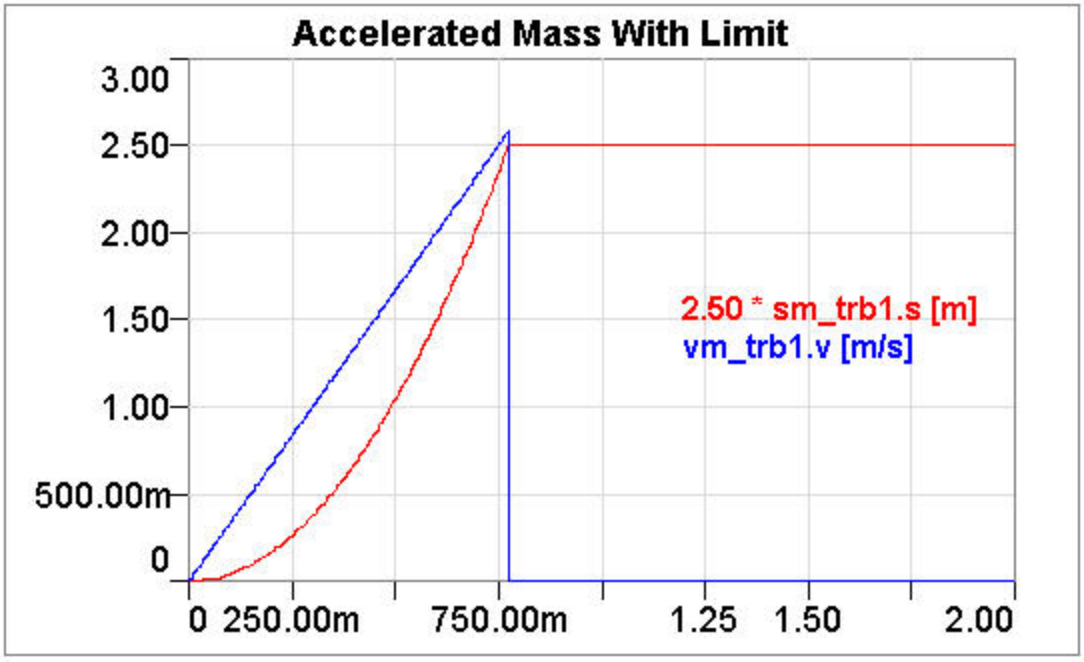


Figure 2. Simulation results-showing velocity and position change up to the limit.

Velocity-Force-Representation

The displacement-force representation consists of:

- Rotational V Components
- Translational V Components

Velocity-Force Rotational V Components

- Damper in VHDL-AMS (damp_rot)
- Torque Source in VHDL-AMS (f_rot)
- Limit Stop in VHDL-AMS (limit_rot)
- Mass in VHDL-AMS (mass_rot)
- Spring in VHDL-AMS (spring_rot)
- Angular Velocity Source in VHDL-AMS (v_rot)

Damper - VHDL-AMS

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

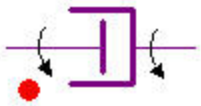


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents a simple damper specified by a linear damping coefficient.

Top

Assumptions and Limitations

Top

Mathematical Description

The Damper model can be described as:

$$m(t) = damping \cdot omega(t)$$

Where m(t) is the torque through terminal rot1 to rot2, and omega(t) is the angular velocity.

Top

Netlist Syntax

COUPL damp_rot ?InstanceName(@InstanceName):(@(Refbase)@(ID)) rot1 := %0 , rot2 := %1 (damping := @damping) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
rot1	Plus Terminal	rotational_velocity
rot2	Minus Terminal	rotational_velocity

Top

Parameters

Table 2

Name	Description	Data Type	Default Value[Unit]
damping	Damping Coefficient	real	1.0 [Nm*s/rad]

Top

Example

See Angular Velocity Source/Spring/Damper Rotational Velocity-Force Example

Top

References

Torque Source - VHDL-AMS

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents an ideal mechanical force/torque source which supplies the defined force/torque for an arbitrary position.

Top

Assumptions and Limitations

The position of an ideal force/torque source can be arbitrary. Real flow sources have always an internal resistance. If the internal resistance is essential for the simulation target, a corresponding resistance must be connected in parallel.

Top

Mathematical Description

Top

Netlist Syntax

COUPL f_rot ?InstanceName(@InstanceName):(@(Refbase)@(ID)) rot1 := %0 , rot2 := %1 (value := @value , ac_mag := @ac_mag , ac_phase := @ac_phase) DST: SIM

(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:-
:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
rot1	Plus Terminal	rotational_velocity
rot2	Minus Terminal	rotational_velocity

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
value	Torque [Nm]	real	1.0 [Nm]
ac_mag	Magnitude of Torque (AC)	real	1m [Nm]
ac_ phase	Phase Shift of Torque (AC)	real	0 [rad]

Top

Example

See Torque Source/Mass/Limit Stop Rotational Velocity-Force Example

Top

References

Limit Stop - VHDL-AMS

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents a parallel damper and spring combination within defined limits (upper and lower limit).

Top

Assumptions and Limitations

Top

Mathematical Description

The Limit Stop can be described as:

If phi(t) > phiul, m(t) = c × [phi(t) - phiul] + damping × omega(t)

If phi(t) < phiul, m(t) = c × [phi(t) – phill] + damping × omega(t)

Else m(t) = 0

where m(t) and phi(t) are the torque through and angle across terminal rot1 and rot2, respectively.

Top

Netlist Syntax

COUPL limit_rot ?InstanceName(@InstanceName):(@(Refbase)@(ID)) rot1 := %0 , rot2 := %1 (phiul := @phiul , phill := @phill , phi0 := @phi0 , c := @c , damping := @damping) DST: SIM (Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:-:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
rot1	Plus Terminal	rotational_velocity
rot2	Minus Terminal	rotational_velocity

Top

Parameters

Table 2

Name	Description	Data Type	Default Value[Unit]
c	Spring Rate	real	1.0e6 [Nm/rad]
damping	Damping Coefficient	real	1.0e6 [Nm*s/rad]
phiul	Upper Angle Limit	real	1.0 [rad]
phill	Lower Angle Limit	real	0.0 [rad]
phi0	Initial Angle	real	0.0 [rad]

Top

Example

See Torque Source/Mass/Limit Stop Rotational Velocity-Force Example

Top

References

Mass - VHDL-AMS

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

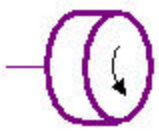


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents a mass specified by a mass/inertia and initial position/velocity.

Initial Position, and Initial Velocity are of the VHDL data type generic, therefore, their numerical values are defined at t = 0 and remain unchanged for the remainder of the simulation.

Top

Assumptions and Limitations

Top

Mathematical Description

The Mass model can be described as:

$$m(t) = j \cdot acc(t)$$

Where m(t) is the torque through terminal rot1 to the mechanical reference, and acc(t) is the angular acceleration.

Top

Netlist Syntax

COUPL mass_rot ?InstanceName(@InstanceName):(@(Refbase)@(ID)) rot1 := %0 (phi0 := @phi0 , omega0 := @omega0 , j := @j) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
rot1	Plus Terminal	rotational_velocity

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
j	Moment of Inertia	real	1.0 [kg*m^2]
phi0	Initial Initial Angle	real	0.0 [rad]
omega0	Initial Angular Velocity	real	0.0 [rad/s]

Top

Example

See Torque Source/Mass/Limit Stop Rotational Velocity-Force Example

Top

References

Spring - VHDL-AMS

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents a simple spring specified by a spring rate.

Top

Assumptions and Limitations

Top

Mathematical Description

The Spring model can be described as:

$$m(t) = c \cdot phi(t)$$

Where m(t) and phi(t) are the torque through and angle across the terminal rot1 and rot2, respectively.

Top

Netlist Syntax

COUPL spring_rot ?InstanceName(@InstanceName):(@(Refbase)@(ID)) rot1 := %0 , rot2 := %1 (phi0 := @phi0 , c := @c) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
rot1	Plus Terminal	rotational_velocity
rot2	Minus Terminal	rotational_velocity

Top

Parameters

Table 2

Name	Description	Data Type	Default Value[Unit]
c	Spring Rate	real	1.0 [Nm/rad]
phi0	Initial Torsional Angle	real	0 [rad]

Top

Example

See Angular Velocity Source/Spring/Damper Rotational Velocity-Force Example

Top

References

Angular Velocity Source - VHDL-AMS

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

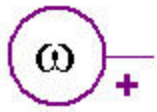


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents an ideal mechanical velocity source which supplies the defined velocity for an arbitrary force/torque.

Top

Assumptions and Limitations

The force/torque through an ideal position/angle source can be arbitrary. Real force/torque sources have always an internal resistance. If the internal resistance is essential for the simulation target, a corresponding resistance must be connected in series.

Top

Mathematical Description

Top

Netlist Syntax

COUPL v_rot ?InstanceName(@InstanceName):(@(Refbase)@(ID)) rot1 := %0 (value := @value , ac_mag := @ac_mag , ac_phase := @ac_phase) DST: SIM(Type:=SimVHDL) SRC:

DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
rot1	Plus Terminal	rotational_velocity

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
value	Angular Velocity	real	1.0 [rad/s]
ac_mag	Magnitude of Angular Velocity (AC)	real	1.0e-3 [rad/s]
ac_phase	Phase Shift of Angular Velocity (AC)	real	0.0 [rad]

Top

Example

See Angular Velocity Source/Spring/Damper Rotational Velocity-Force Example

Top

References

Angular Velocity Source/Spring/Damper Rotational Velocity-Force Example

This example is an underdamped spring/mass system. It combines an angular velocity source, spring, damper and mass model into a single example. In this example, a constant angular velocity is applied to the system. The damping causes the force applied to the mass element to decay sinusoidally over time.

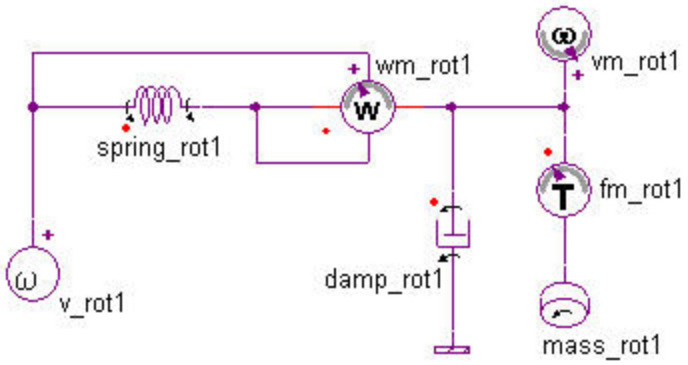


Figure 1. Application examples of the VHDL-AMS Rotational VF Angular Velocity Source with Spring and Damper

Table 1. System Parameters

Component	Parameter	Value [unit]
Angular Velocity Source(Rotational VF) v_rot1	value	10 [rad/s]
Spring (Rotational VF) spring_rot1	c	10 [Nm/rad]
Damper (Rotational VF) damp_rot1	damping	0.05 [Ns/m]
Mass (Rotational VF) mass_rot1	phi0	0.0 [rad]
	omega0	0.0 [rad/s]
	j	1.0 [kg*m%]

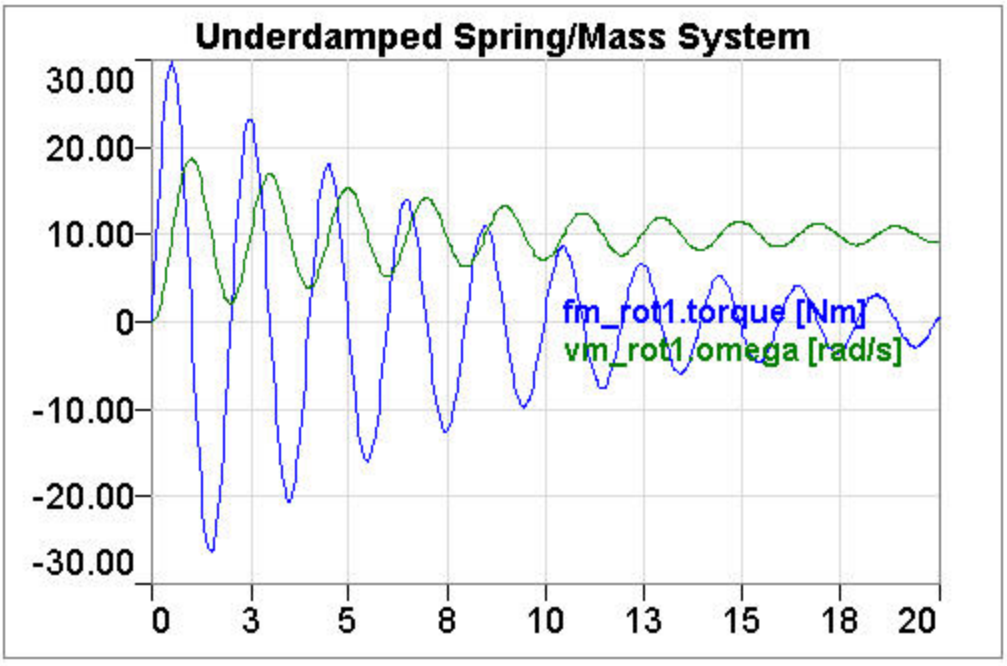


Figure 3. Simulation results-showing velocity and force delivered to mass decaying with time.

Torque Source/Mass/Limit Stop Rotational Velocity-Force Example

This example is a mass accelerated by a constant torque. The acceleration is limited due to the Limit Stop. The Limit Stop represents a spring-damper combination which causes the mass to stop at the defined upper angle limit.

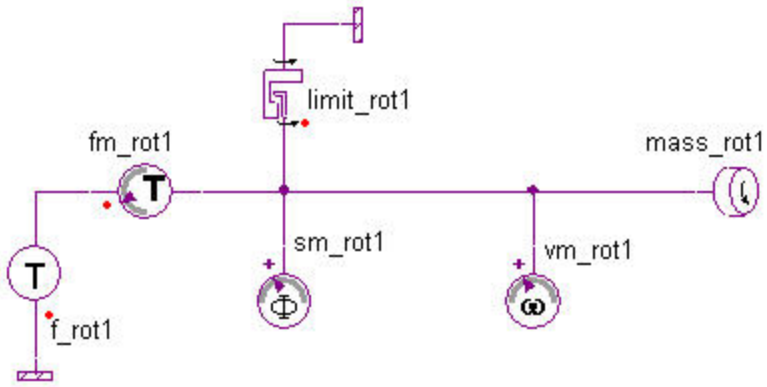


Figure 1. Application examples of the VHDL-AMS Rotational VF Torque Source with Mass and Limit Stop

Table 1. System Parameters

Component	Parameter	Value [unit]
Torque Source(Rotational VF) f_rot1	value	10 [Nm]
Limit Stop (Rotational VF) limit_rot1	c	1e5 [N/m]
	phiul	1.0
	damping	1e5 [Ns/m]
	phill	0
Mass (Rotational VF) mass_rot1	phi0	0.0 [rad]
	omega0	0.0 [rad/s]
	j	3 [kg*m%]

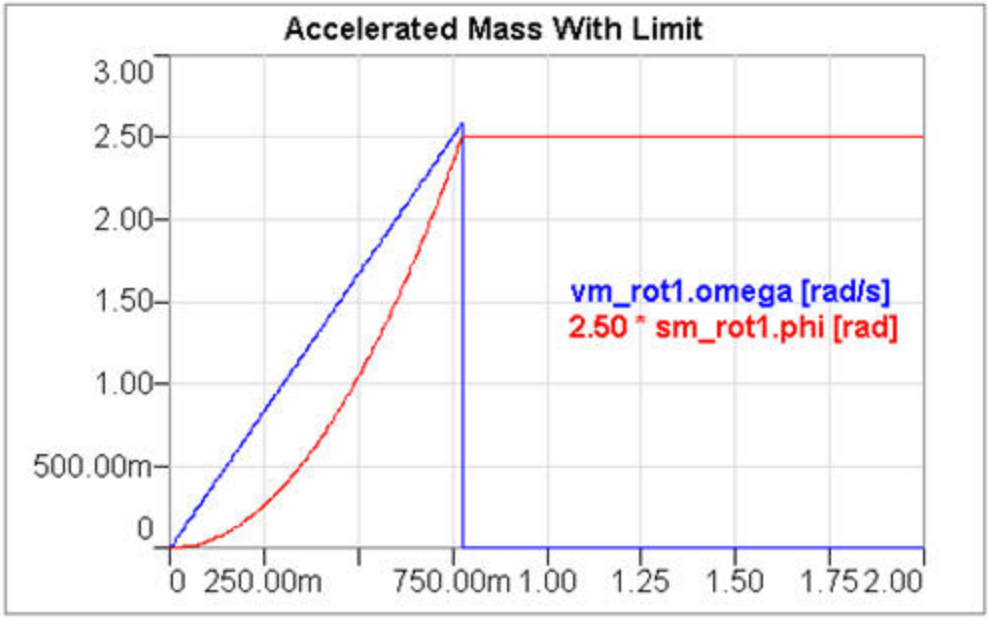


Figure 3. Simulation results-showing velocity and position change up to the limit, as well as, the constant input force.

Velocity-Force Translational V Components

- Damper in VHDL-AMS (damp_tr)
- Force Source in VHDL-AMS (f_tr)
- Limit Stop in VHDL-AMS (limit_tr)
- Mass in VHDL-AMS (mass_tr)
- Spring in VHDL-AMS (spring_tr)
- Velocity Source in VHDL-AMS (v_tr)

Damper - VHDL-AMS

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

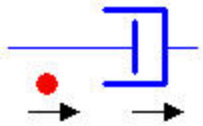


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents a simple damper specified by a linear damping coefficient.

Top

Assumptions and Limitations

Top

Mathematical Description

The Damper model can be described as:

$$F(t) = damping \cdot v(t)$$

Where F(t) is the force through terminal tr1 to tr2, and v(t) is the velocity.

Top

Netlist Syntax

COUPL damp_tr ?InstanceName(@InstanceName):(@(Refbase)@(ID)) tr1 := %0 , tr2 := %1 (damping := @damping) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
tr1	Plus Terminal	translational_velocity
tr2	Minus Terminal	translational_velocity

Top

Parameters

Table 2

Name	Description	Data Type	Default Value[Unit]
damping	Damping Coefficient	real	1.0 [Ns/m]

Top

Example

See Velocity Source/Spring/Damper Translational Velocity-Force Example

Top

References

Force Source - VHDL-AMS

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

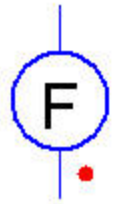


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents an ideal mechanical force/torque source which supplies the defined force/torque for an arbitrary position.

Top

Assumptions and Limitations

The position of an ideal force/torque source can be arbitrary. Real flow sources have always an internal resistance. If the internal resistance is essential for the simulation target, a corresponding resistance must be connected in parallel.

Top

Mathematical Description

Top

Netlist Syntax

COUPL f_tr ?InstanceName(@InstanceName):(@(Refbase)@(ID)) tr1 := %0 , tr2 := %1 (value := @value , ac_mag := @ac_mag , ac_phase := @ac_phase) DST: SIM(Type:=SimVHDL)

SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
tr1	Plus Terminal	translational_velocity
tr2	Minus Terminal	translational_velocity

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
value	Force	real	1.0 [N]
ac_mag	Magnitude of Force (AC)	real	1m [N]
ac_ phase	Phase Shift of Force (AC)	real	0 [rad]

Top

Example

See Force Source/Mass/Limit Stop Translational Velocity-Force Example

Top

References

Limit Stop - VHDL-AMS

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

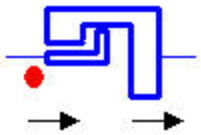


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents a parallel damper and spring combination within defined limits (upper and lower limit).

Top

Assumptions and Limitations

Top

Mathematical Description

The Limit Stop can be described as:

If s(t) > sul, F(t) = c × [s(t) - sul] + damping × v(t)

If s(t) < sul, F(t) = c × [s(t) – sll_val] + damping × v(t)

Else F(t) = 0

where F(t) and s(t) are the force through and displacement across terminal tr1 and tr2.

Top

Netlist Syntax

COUPL limit_tr ?InstanceName(@InstanceName):(@(Refbase)@(ID)) tr1 := %0 , tr2 := %1 (sul := @sul , sll_val := @sll_val , s0 := @s0 , c := @c , damping := @damping) DST: SIM(Type:-:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
tr1	Plus Terminal	translational_velocity
tr2	Minus Terminal	translational_velocity

Top

Parameters

Table 2

Name	Description	Data Type	Default Value[Unit]
c	Spring Rate	real	1.0e6 [N/m]
damping	Damping Coefficient	real	1.0e6 [Ns/m]
sul	Upper Position Limit	real	1.0 [m]
sll_val	Lower Position Limit	real	0.0 [m]
s0	Initial Position	real	0.0 [m]

Top

Example

See Force Source/Mass/Limit Stop Translational Velocity-Force Example

Top

References

Mass - VHDL-AMS

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents a mass specified by a mass/inertia and initial position/velocity.

Initial Position, and Initial Velocity are of the VHDL data type generic, therefore, their numerical values are defined at t = 0 and remain unchanged for the remainder of the simulation.

Top

Assumptions and Limitations

Top

Mathematical Description

The Mass model can be described as:

$$F(t) = m \cdot a(t)$$

Where f(t) is the force through the terminal tr1 to mechanical reference, and a(t) is the acceleration of the mass.

Top

Netlist Syntax

COUPL mass_tr ?InstanceName(@InstanceName):(@(Refbase)@(ID)) tr1 := %0 (s0 := @s0 , v0 := @v0 , m := @m) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:-:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
tr1	Plus Terminal	translational_velocity

Top

Parameters

Table 2

Name	Description	Data Type	Default Value[Unit]
m	Mass	real	1.0 [kg]
s0	Initial Displacement	real	0.0 [m]
v0	Initial Velocity	real	0.0 [m/s]

Top

Example

See Force Source/Mass/Limit Stop Translational Velocity-Force Example

Top

References

Spring - VHDL-AMS

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

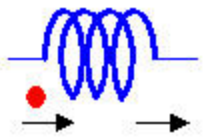


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents a simple spring specified by a spring rate.

Top

Assumptions and Limitations

Top

Mathematical Description

The Spring model can be described as:

$$f(t) = c \cdot s(t)$$

Where f(t) and s(t) are the force through and the displacement across the terminal tr1 and tr2.

Top

Netlist Syntax

COUPL spring_tr ?InstanceName(@InstanceName):(@(Refbase)@(ID)) tr1 := %0 , tr2 := %1 (s0 := @s0 , c := @c) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:-:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
tr1	Plus Terminal	translational_velocity
tr2	Minus Terminal	translational_velocity

Top

Parameters

Table 2

Name	Description	Data Type	Default Value[Unit]
c	Spring Rate	real	100.0 N/m
s0	Initial Displacement	real	0 [m]

Top

Example

See Velocity Source/Spring/Damper Translational Velocity-Force Example

Top

References

Velocity Source - VHDL-AMS

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

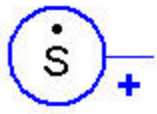


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents an ideal mechanical velocity source which supplies the defined velocity for an arbitrary force/torque.

Top

Assumptions and Limitations

The force/torque through an ideal position/angle source can be arbitrary. Real force/torque sources have always an internal resistance. If the internal resistance is essential for the simulation target, a corresponding resistance must be connected in series.

Top

Mathematical Description

Top

Netlist Syntax

COUPL v_tr ?InstanceName(@InstanceName):(@(Refbase)@(ID)) tr1 := %0 (value := @value , ac_mag := @ac_mag , ac_phase := @ac_phase) DST: SIM(Type:=SimVHDL) SRC: DB

(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
tr1	Plus Terminal	translational_velocity

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
value	Velocity [m/s]	real	1.0 [m/s]
ac_mag	Magnitude of Velocity (AC)	real	1.0e-3 [m]
ac_ phase	Phase Shift of Velocity (AC)	real	0.0

Top

Example

See Velocity Source/Spring/Damper Translational Velocity-Force Example

Top

References

Velocity Source/Spring/Damper Translational Velocity-Force Example

This example is an underdamped spring/mass system. It combines a velocity source, spring, damper and mass model into a single example. In this example, a constant velocity is applied to the system. The damping causes the force applied to the mass element to decay sinusoidally over time.

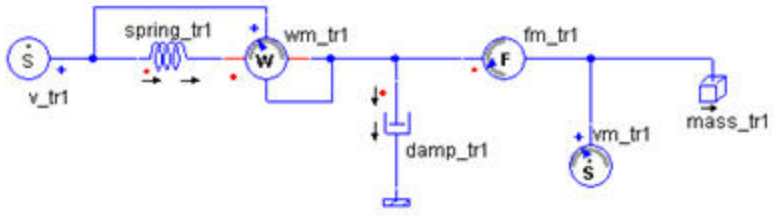


Figure 1. Application examples of the VHDL-AMS Translational VF Velocity Source with Spring and Damper

Table 1. System Parameters

Component	Parameter	Value [unit]
Velocity Source(Translational VF) v_tr1	value	10 [m/s]
Spring (Translational VF) spring_tr1	c	0.1 [N/m]
Damper (Translational VF) damp_tr1	damping	0.05 [Ns/m]
Mass (Translational VF) mass_tr1	s0	0.0 [m]
	v0	0.0 [m/s]
	m	1.0 [kg]

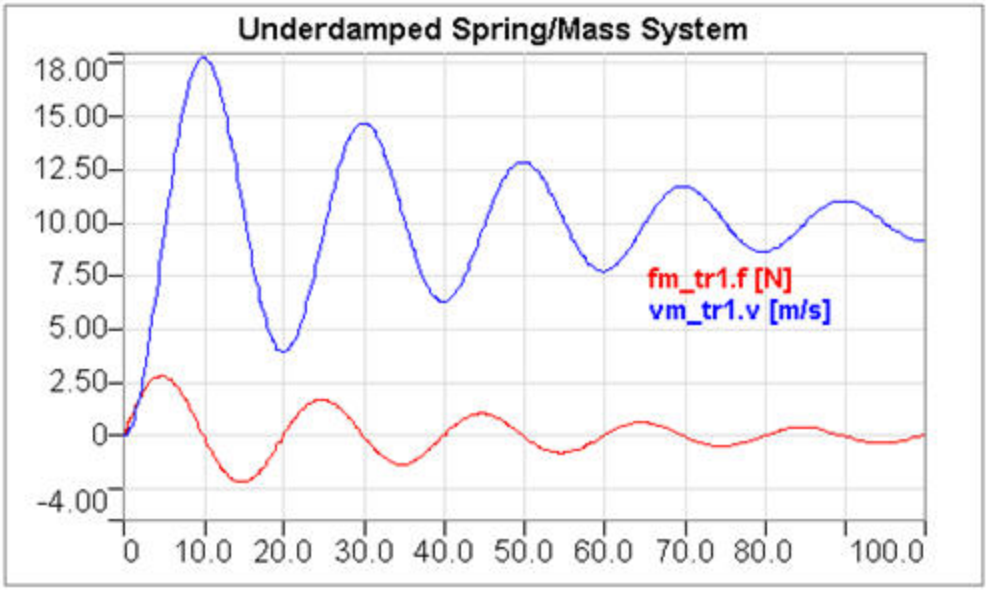


Figure 2. Simulation results-showing velocity and force delivered to mass.

Force Source/Mass/Limit Stop Translational Velocity-Force Example

This example is a mass accelerated by a constant force. The acceleration is limited due to the Limit Stop. The Limit Stop represents a spring-damper combination which causes the mass to stop at the defined upper limit.

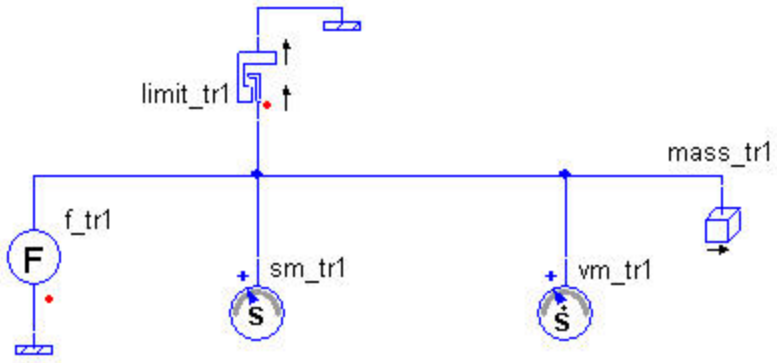


Figure 1. Application examples of the VHDL-AMS Translational VF Force Source with Mass and Limit Stop

Table 1. System Parameters

Component	Parameter	Value [unit]
Force Source(Translational VF) f_tr1	value	10 [N]
Limit Stop (Translational VF) limit_tr1	c	1e6 [N/m]
	sll_val	0
	damping	1e6 [Ns/m]
	sul	4
Mass (Translational VF) mass_tr1	s0	0.0 [m]
	v0	0.0 [m/s]
	m	3 [kg]

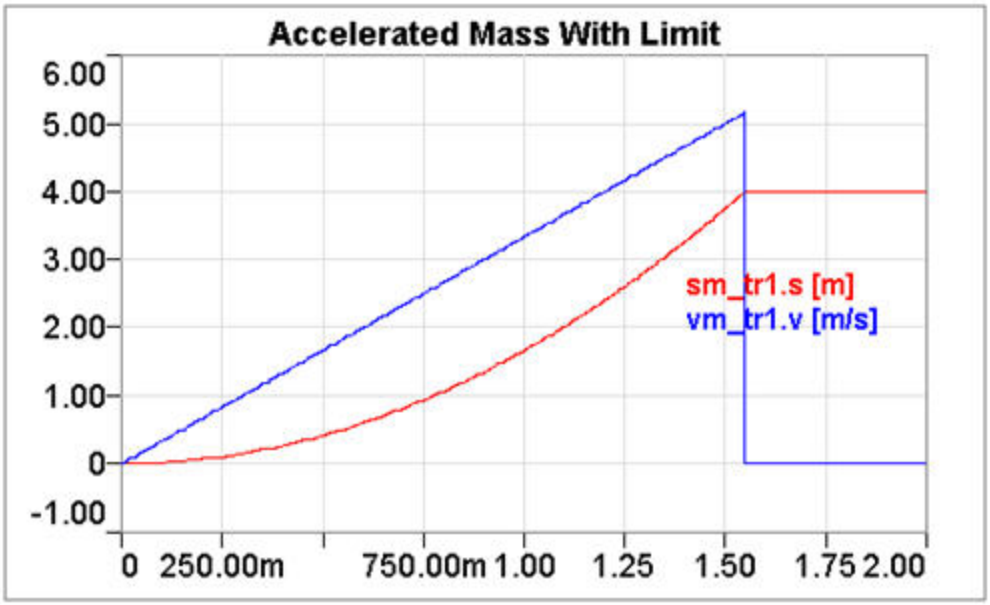


Figure 2. Simulation results-showing velocity and position change up to the limit.

Thermal Components - VHDL-AMS

• Thermal Capacitance in VHDL-AMS (cth)

• Heat Flow Source in VHDL-AMS (h)

• Thermal Resistance in VHDL-AMS (rth)

• Temperature Source in VHDL-AMS (t)

The thermal domain components appear in the *Thermal* folder of *Physical Domains* in the **Basic Element VHDLAMS** library. The folder provides thermal basic components to model simple thermal equivalent circuits, for example cooling systems or heat sinks. The thermal domain uses the temperature as across and the heat flow as through quantity.

By default, components and wires of the thermal domain are represented in red in the graphical representation of the Schematic. To change the settings of nature colors, choose **Tools>Options>Schematic Editor Options**, click on the Colors tab, select Wires as the object type, and the color of wires for different domains can be edited. Please note: You can only change the wiring color, not the color of the component.

Across and Through Quantity of the Thermal Domain

Across	Through	Equation
Temperature [K]	Heat Flow [J/s]	t = K * h

Thermal Capacitance - VHDL-AMS

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

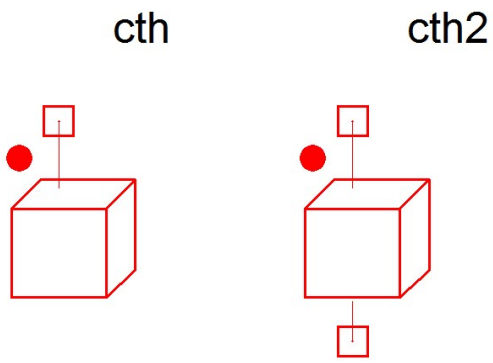


Figure 1. Component symbol

• Description
• Assumptions and Limitations
• Mathematical Description
• Netlist Syntax
• Conservative Pins
• Parameters
• Example
• References

Description

The component represents an ideal thermal capacitance.

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

For single-pin thermal capacitance cth:

COUPL cth ?InstanceName(@InstanceName):(@(Refbase)@(ID)) th1 := %0 (c_th := @c_th , t0 := @t0) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\");

For double-pin thermal capacitance cth2:

COUPL cth2 ?InstanceName(@InstanceName):(@(Refbase)@(ID)) th1 := %0 (c_th := @c_th , t0 := @t0) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\");

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
th1	Plus Terminal	thermal
th2 (for cth2 only)	Minus Terminal	thermal

Top

Parameters

Table 2: Parameters of single-pin thermal capacitance cth

Name	Description	Data Type	Default Value [Unit]
c_th	Thermal Capacitance	Thermal_Capacitance	1.0 [J/K]
t0	Initial Temperature	temperature	293.0 [K]

Table 3: Parameters of double-pin thermal capacitance cth2

Name	Description	Data Type	Default Value [Unit]
c_th	Thermal Capacitance	Thermal_Capacitance	1.0 [J/K]
t0	Initial Temperature Difference Between Pin th1 and th2	Temperature Difference	0 [delta_K]

Top

Example

See Thermal Component Example

Top

References

Heat Flow Source - VHDL-AMS

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents an ideal flow source which supplies the defined flow rate for an arbitrary pressure differential.

The pressure of an ideal flow source can be arbitrary. Real flow sources have always an internal resistance. If the internal resistance is essential for the simulation target, a corresponding resistance must be connected in parallel.

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

COUPL h ?InstanceName(@InstanceName):(@(Refbase)@(ID)) th1 := %0 , th2 := %1 (value := @value , ac_mag := @ac_mag , ac_phase := @ac_phase) DST: SIM(Type:=SimVHDL)

SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
th1	Plus Terminal	thermal
th2	Minus Terminal	thermal

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
value	Heat Flow	real	1.0 [J/s]
ac_mag	Magnitude of Heat Flow (AC)	real	1.0e-3
ac_ phase	Phase Shift of Heat Flow (AC)	real	0.0

Top

Example

See Thermal Component Example

Top

References

Thermal Resistance - VHDL-AMS

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents a ideal thermal resistance specified ba a linear coefficient.

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

COUPL rth ?InstanceName(@InstanceName):(@(Refbase)@(ID)) th1 := %0 , th2 := %1 (k := @k) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:-:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
th1	Plus Terminal	thermal
th2	Minus Terminal	thermal

Top

Parameters

Table 2

Name	Description	Data Type	Default Value[Unit]
k	Linear Coefficient	real	1.0 [K*s/J]

Top

Example

See Thermal Component Example

Top

References

Temperature Source - VHDL-AMS

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

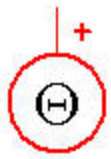


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Conservative Pins

• Parameters

• Example

• References

Description

The component represents an ideal temperature source which supplies the defined temperature for an arbitrary heat flow.

The heat flow through an ideal temperature source can be arbitrary. Real temperature sources have always an internal resistance. If the internal resistance is essential for the simulation target, a corresponding resistance must be connected in series.

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

COUPL t ?InstanceName(@InstanceName):(@(Refbase)@(ID)) th1 := %0 (value := @value , ac_mag := @ac_mag , ac_phase := @ac_phase) DST: SIM(Type:=SimVHDL) SRC: DB

(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Conservative Pins

Table 1

Name	Port/Terminal Description	Nature/Data Type
th1	Plus Terminal	thermal

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
value	Temperature	real	293.0 [K]
ac_mag	Magnitude of Temperature (AC)	real	1.0e-3 [K]
ac_ phase	Phase Shift of Temperature (AC)	real	0.0

Top

Example

See Thermal Component Example

Top

References

Thermal Component Example

This example illustrates the charging and discharging of a thermal capacitance. In one case a temperature source is used, and in the second a heat flow source is used to charge the capacitor. The transient behavior of the capacitor is monitored by a thermometer.

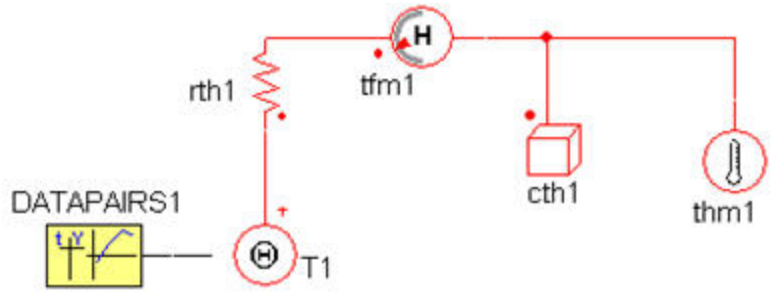


Figure 1. Application examples of the VHDL-AMS Temperature Source

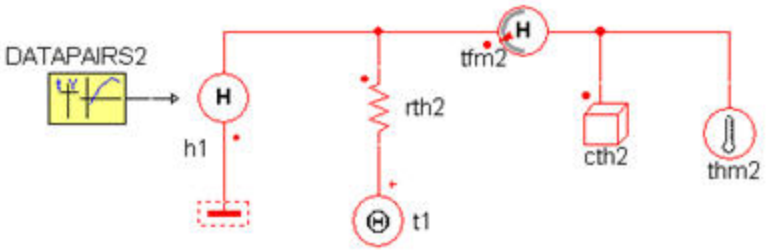


Figure 2. Application examples of the VHDL-AMS Heat Flow Source

Table 1. System Parameters

Component	Parameter	Value [unit]
2D Lookup Table Element DATAPAIRS1	tperio	20 [S]
	tdelay	0 [S]
	perio	1
	file	thermal_SSH__ DATAPAIRS1.mdx
2D Lookup Table Element	tperio	20 [S]

DATAPAIRS1		
	tdelay	0 [S]
	perio	1
	file	thermal_SSH__ DATAPAIRS2.mdx
Temperature Source T1	value	DATAPAIRS1.val
Temperature Source t1	value	273 [K]
Thermal Resistance rth1/rth2	k	0.25 [K/W]
Thermal Capacitance cth1	c_th	1 [J/K]
	t0	273 [K]
Thermal Capacitance cth2	c_th	1 [J/K]
	t0	313 [K]
Heat Flow Source h1	value	DATAPAIRS2.val

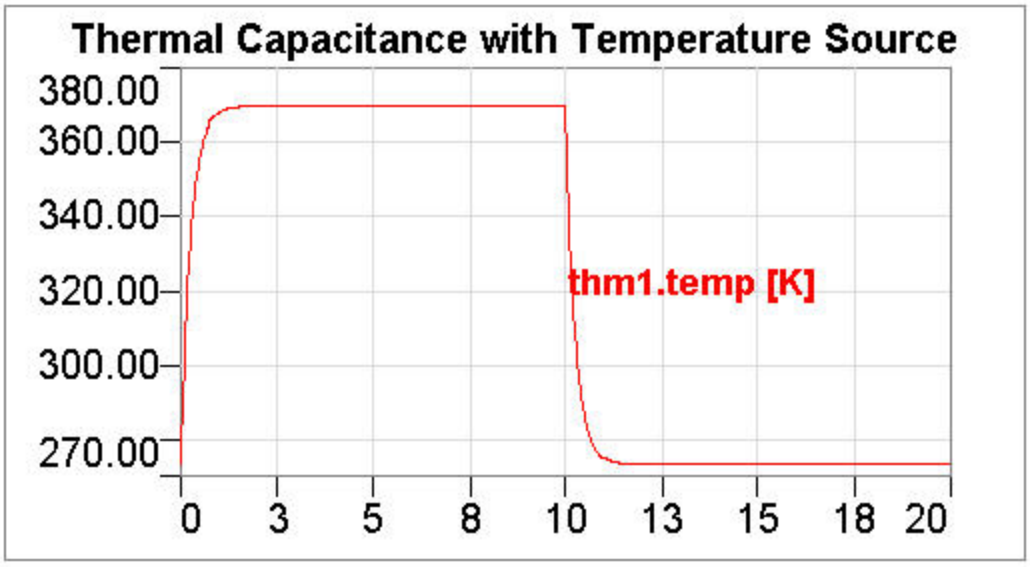


Figure 3. Simulation results-thermal capacitance temperature vs time with temperature source.

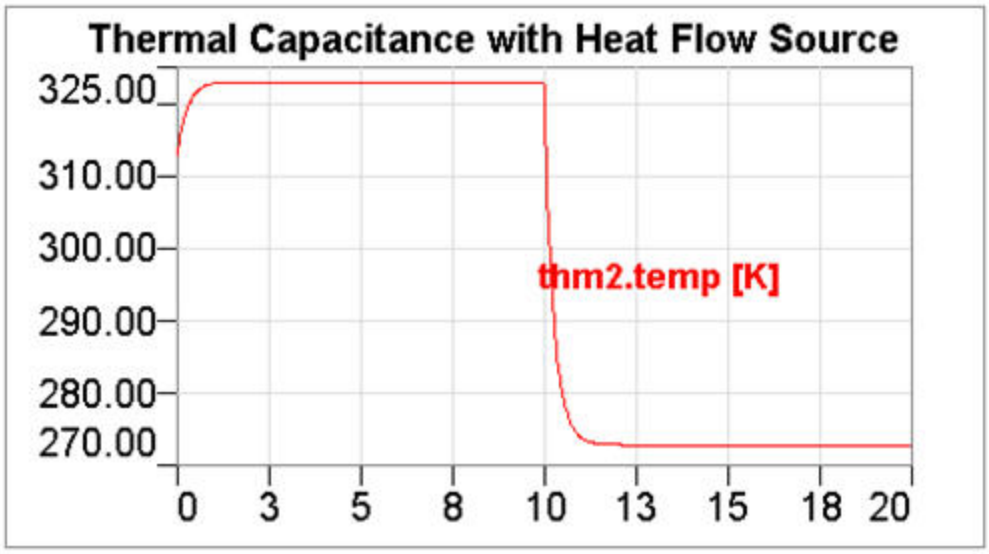


Figure 4. Simulation results-thermal capacitance temperature vs time with heat flow source.

Modeling Tools

Complex simulation models usually need simple auxiliary functions to model typical time functions. So the real subject of simulation can be investigated faster and more precisely.

VHDL-AMS modeling tools provide various time functions. The components appear in the **Basic Element VHDLAMS** library. Open the Tools folder, choose *Time Functions,* click a name, drag the component onto the sheet and release the mouse button.

See also Using Time Functions.

Using Time Functions

Time functions can be used for modeling in several ways.

• In electrical circuits, time functions can alter transient input signals of externally controlled sources.

• In block diagrams, time functions can define nominal values, disturbances and reference values.

• In state graphs time functions define comparison functions and synchronization signals.

Predefined time functions have a quality determined behavior for which certain parameters can be defined. Predefined time functions are: Sine, Pulse, Triangular, Sawtooth, and Trapezoidal wave; as well Needle, pulse, and Arc tangent.

In the property dialog box, the frequency, period, initial delay, amplitude, phase, periodicity and offset can be modified.

Time Functions

- Arc Tan Time Function in VHDL-AMS (arctan)
- Needle Time Function in VHDL-AMS (needle)
- Pulse Wave Time Function in VHDL-AMS (pulse)
- Sawtooth Wave Time Function in VHDL-AMS (sawtoothl)
- Sawtooth Wave (Rising) Time Function in VHDL-AMS (sawtoothr)
- Sine Wave Time Function in VHDL-AMS (sine)
- Trapezoidal Wave Time Function in VHDL-AMS (trapez)
- Triangular Wave Time Function in VHDL-AMS (triang)

Arc Tangent

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

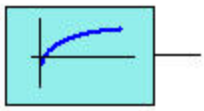


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Parameters

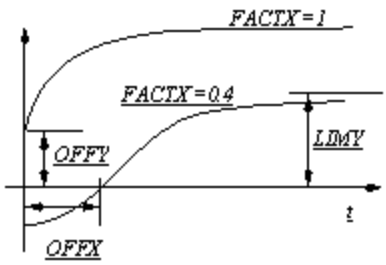
• Input/Output Quantities

• Example

• References

Description

This model generates an arc tangent waveform as a function of the simulation time.



Top

Assumptions and Limitations

Top

Mathematical Description

The output of the Arc Tangent Function can be described as following:

$$val(t) = \frac{\lim\ y}{\pi / 2} \cdot \arctan[\ factx \cdot (t - offx)] + offy$$

Top

Netlist Syntax

COUPL arctan ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (limy := @limy , factx := @factx , offx := @offx , offy := @offy) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value[Unit]
limy	Limit Y	real	1.578
factx	Factor X	real	1.0
offx	Offset X	real	0.0
offy	Offset Y	real	0.0

Top

Input/Output Quantities

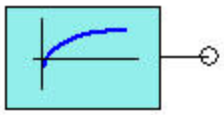
Table 2

Name	Description [Unit]	Direction	Data Type
val	Value	Output	real

Top

Example

This example illustrates the output of Arc Tangent Time Function Block.



arctan1

Figure 2. Application example of the VHDL-AMS Arc Tangent Time Function Block

Table 3. System Parameters

Component	Parameter	Value [unit]
Arc Tangent Time Function Block arctan1	limy	1.578
	factx	3
	offx	0.5
	offy	2

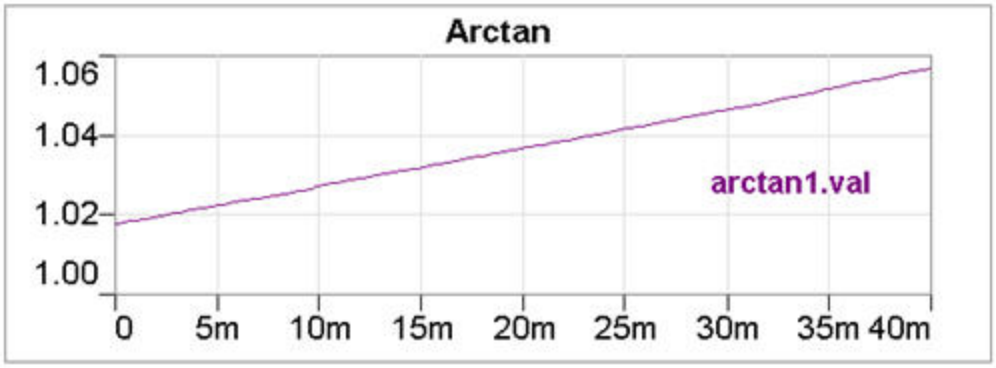


Figure 3. Simulation results-output from the Arc Tangent Function block.

Top

References

Needle

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

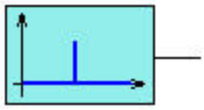


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Parameters

• Input/Output Quantities

• Example

• References

Description

The function provides needle pulses with a defined amplitude and frequency after a delay to the simulation start and a delay within the period at the output.

The ts value for the model has to always be hmin (to get the needle effect). The sample time cannot be specified as '0'. If a '0' value is specified, the model provides an error message.

Top

Assumptions and Limitations

Top

Mathematical Description

The output of the Needle Pulses model can be described as following:

$$val(t) = ampl \quad if\ t = tdelay + t0 + \frac{k}{freq} \quad where\ k \in N$$

Top

Netlist Syntax

COUPL needle ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (ampl := @ampl , freq := @freq , off := @off , t0 := @t0 , ts := @ts , tdelay := @tdelay) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value[Unit]
freq	Frequency	real	50.0 [Hz]
ampl	Amplitude of Pulses	real	1.0
t0	Delay within the Period	real	0.0 [s]
ts	Sample Time	real	0.0 [s]
off	Offset	real	0.0
tdelay	Initial Delay	real	0.0 [s]

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Value	Output	real

Top

Example

This example illustrates the output of Needle Time Function Block.

needle1

Figure 2. Application example of the VHDL-AMS Needle Time Function Block

Table 3. System Parameters

Component	Parameter	Value [unit]
Needle Time Function Block needle1	ts	hmin
	tdelay	0 [S]
	t0	0 [S]
	off	0
	freq	50 [Hz]
	ampl	326

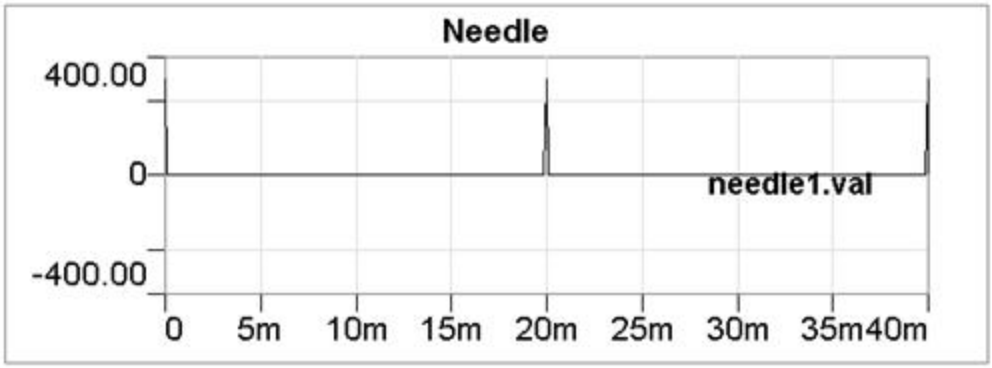


Figure 3. Simulation results-output from the Needle Function block.

Top

References

Pulse

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

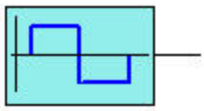


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Parameters

• Input/Output Quantities

• Example

• References

Description

This model generates a pulse waveform as a function of the simulation time.

Top

Assumptions and Limitations

Top

Mathematical Description

The output of the Pulse Wave can be described as following:

$$val(t) = ampl \cdot pulse[360^{o} \cdot freq \cdot (t - tdelay) + phase] + off$$

Top

Netlist Syntax

COUPL pulse ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (ampl := @ampl , freq := @freq , off := @off , tdelay := @tdelay) DST: SIM(Type:=SimVHDL) SRC: DB

(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value[Unit]
freq	Frequency	real	50.0 [Hz]
ampl	Amplitude	real	326.0
off	Offset	real	0.0
tdelay	Initial Delay	real	0.0 [s]

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Value	Output	real

Top

Example

This example illustrates the output of Pulse Wave Time Function Block.

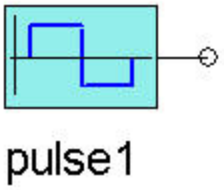


Figure 2. Application example of the VHDL-AMS Pulse Wave Time Function Block

Table 3. System Parameters

Component	Parameter	Value [unit]
Pulse Wave Time Function Block pulse1	off	0 [S]
	tdelay	0 [S]

	freq	50 [Hz]
	ampl	326

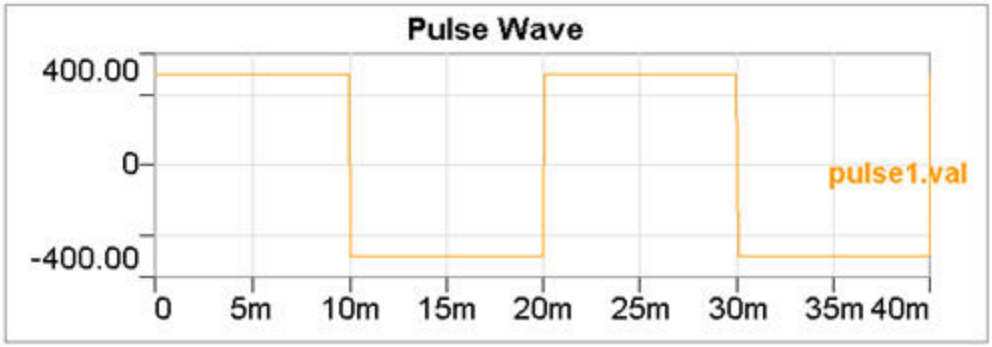


Figure 3. Simulation results-output from the Pulse Wave block.

Top

References

Sawtooth Wave Falling

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

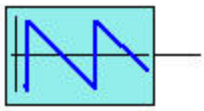


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Parameters

• Input/Output Quantities

• Example

• References

Description

This model generates a falling saw-tooth waveform as a function of the simulation time.

Top

Assumptions and Limitations

Top

Mathematical Description

The output of the Saw-Tooth Function (falling) can be described as following:

$$val(t) = ampl \cdot sawtooh_{falling}\ [360^{o} \cdot freq \cdot (t - tdelay) + phase) + off$$

Top

Netlist Syntax

COUPL sawtoothl ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (ampl := @ampl , freq := @freq , off := @off , tdelay := @tdelay) DST: SIM(Type:=SimVHDL) SRC: DB

(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value[Unit]
freq	Frequency	real	50.0 [Hz]
ampl	Amplitude	real	326.0
off	Offset	real	0.0
tdelay	Initial Delay	real	0.0 [s]

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Value	Output	real

Top

Example

This example illustrates the output of Saw-Tooth Wave Falling Time Function Block.

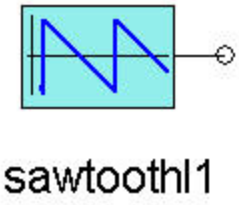


Figure 2. Application example of the VHDL-AMS Saw-Tooth Wave Falling Time Function Block

Table 3. System Parameters

Component	Parameter	Value [unit]
Sawtooth Wave Falling Time Function Block sawtoothl1	off	0 [S]

	tdelay	0 [S]
	freq	50 [Hz]
	ampl	326

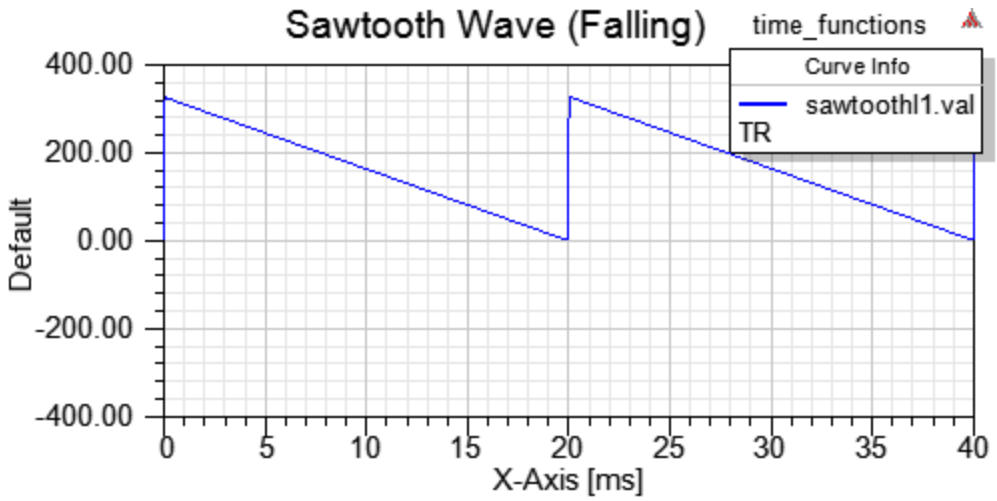


Figure 3. Simulation results-output from the Saw-Tooth Wave (falling) block.

Top

References

Sawtooth Wave Rising

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

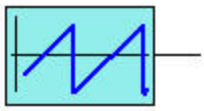


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Parameters

• Input/Output Quantities

• Example

• References

Description

This model generates a rising saw-tooth waveform as a function of the simulation time.

Top

Assumptions and Limitations

Top

Mathematical Description

The output of the Saw-Tooth Function (rising) can be described as following:

$$val(t) = ampl \cdot sawtooh_{rising}[360^{o} \cdot freq \cdot (t - tdelay) + phase) + off$$

Top

Netlist Syntax

COUPL sawtoothr ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (ampl := @ampl , freq := @freq , off := @off , tdelay := @tdelay) DST: SIM(Type:=SimVHDL) SRC: DB

(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value[Unit]
freq	Frequency	real	50.0 [Hz]
ampl	Amplitude	real	326.0
off	Offset	real	0.0
tdelay	Initial Delay	real	0.0 [s]

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Value	Output	real

Top

Example

This example illustrates the output of Saw-Tooth Wave Rising Time Function Block.

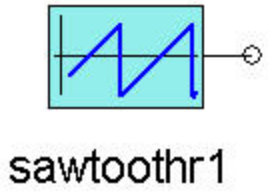


Figure 2. Application example of the VHDL-AMS Saw-Tooth Wave Rising Time Function Block

Table 3. System Parameters

Component	Parameter	Value [unit]
Sawtooth Wave Rising Time Function Block sawtoothr1	off	0 [S]

	tdelay	0 [S]
	freq	50 [Hz]
	ampl	326

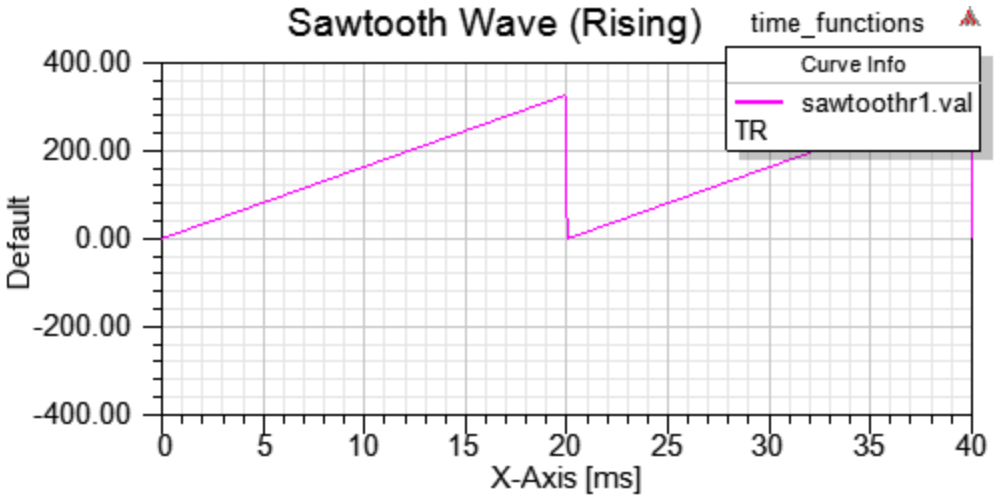


Figure 3. Simulation results-output from the Saw-Tooth Wave (Rising) block.

Top

References

Sine Wave

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

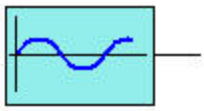


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Parameters

• Input/Output Quantities

• Example

• References

Description

This model generates a sinusoidal waveform as a function of the simulation time.

Top

Assumptions and Limitations

Top

Mathematical Description

The output of the Sine Wave can be described as following:

$$val(t) = ampl \cdot \sin[360^{o} \cdot freq \cdot (t - tdelay) + phase] + off$$

Top

Netlist Syntax

COUPL sine ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (phase := @phase , off := @off , tdelay := @tdelay , ampl := @ampl , freq := @freq) DST: SIM(Type:=SimVHDL) SRC: DB

(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value[Unit]
freq	Frequency	real	50.0 [Hz]
ampl	Amplitude	real	326.0
phase	Phase	real	0.0 [rad]
off	Offset	real	0.0
tdelay	Initial Delay	real	0.0 [s]

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Value	Output	real

Top

Example

This example illustrates the output of Sine Wave Time Function Block.

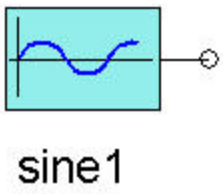


Figure 2. Application example of the VHDL-AMS Sine Wave Time Function Block

Table 3. System Parameters

Component	Parameter	Value [unit]

Sine Wave Time Function Block sine1	off	0 [S]
	tdelay	0 [S]
	freq	50 [Hz]
	ampl	326

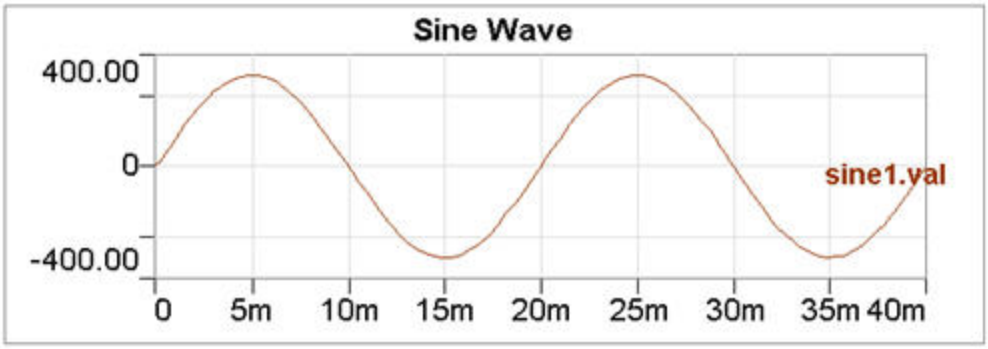


Figure 3. Simulation results-output from the Sine Wave block.

Top

References

Trapezoidal Wave

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

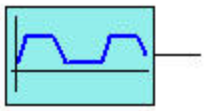


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Parameters

• Input/Output Quantities

• Example

• References

Description

This model generates a trapezoidal waveform as a function of the simulation time.

Top

Assumptions and Limitations

Top

Mathematical Description

This model generates a trapezoidal waveform as a function of the simulation time.

$$val(t) = ampl \cdot trapez[360^{o} \cdot freq \cdot (t - tdelay) + phase) + off$$

$$where\ freq = \frac{1}{trise + tfall + pwidth}$$

Top

Netlist Syntax

COUPL trapez ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (ampl := @ampl , pwidth := @pwidth , off := @off , trise := @trise , tfall := @tfall , tdelay := @tdelay) DST: SIM(Type:-:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value[Unit]
trise	Rise Time	real	5.0e-3 [s]
tfall	Fall Time	real	10.0e-3 [s]
pwidth	Pulse Width	real	5.0e-3 [s]
ampl	Amplitude	real	326.0
off	Offset	real	0.0
tdelay	Initial Delay	real	0.0 [s]

Top

Input/Output Quantities

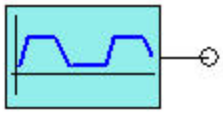
Table 2

Name	Description [Unit]	Direction	Data Type
val	Value	Output	real

Top

Example

This example illustrates the output of Trapezoidal Wave Time Function Block.



trapez1

Figure 2. Application example of the VHDL-AMS Trapezoidal Wave Time Function Block

Table 3. System Parameters

Component	Parameter	Value [unit]
Trapezoidal Wave Time Function Block trapez1	off	0 [S]
	tdelay	0 [S]
	pwidth	5m [S]
	ampl	326
	trise	5m [S]
	tfall	10m [S]

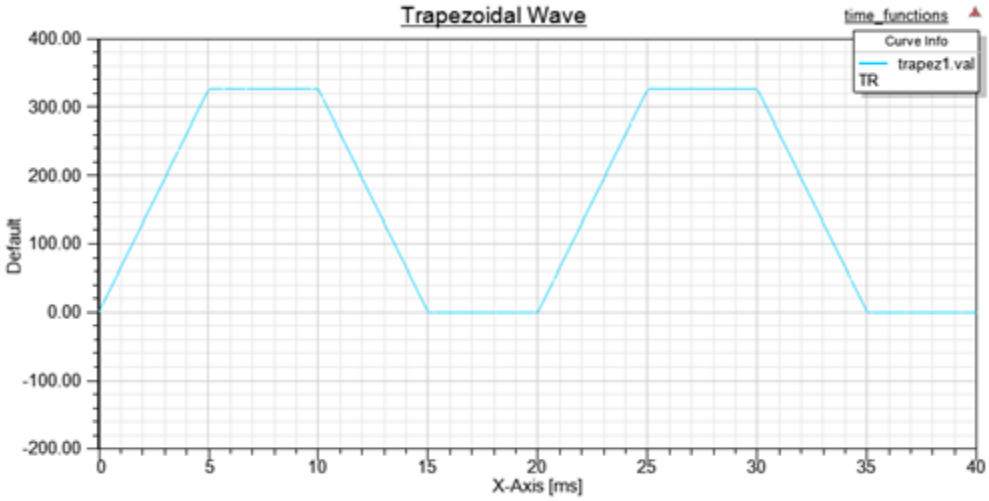


Figure 3. Simulation results-output from the Trapezoidal Wave block.

Top

References

Triangular Wave

Library: Basic Elements VHDLAMS	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

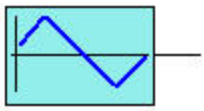


Figure 1. Component symbol

• Description

• Assumptions and Limitations

• Mathematical Description

• Netlist Syntax

• Parameters

• Input/Output Quantities

• Example

• References

Description

This model generates a triangular waveform as a function of the simulation time.

Note: This model gives the same wave form of triangle component as in the Triangular Wave in the Basic Element Library with phase of -90 degree.

Top

Assumptions and Limitations

Top

Mathematical Description

This model generates a triangular waveform as a function of the simulation time.

$$val(t) = ampl \cdot triang[360^{0} \cdot freq \cdot (t - tdelay + phase] + off$$

Top

Netlist Syntax

COUPL triang ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (ampl := @ampl , freq := @freq , off := @off , tdelay := @tdelay) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value[Unit]
freq	Frequency	real	50.0 [Hz]
ampl	Amplitude	real	326.0
off	Offset	real	0.0
tdelay	Initial Delay	real	0.0 [s]
phase	phase	real	0.0 [degree]

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Value	Output	real

Top

Example

This example illustrates the output of Triangular Wave Time Function Block.

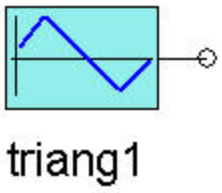


Figure 2. Application example of the VHDL-AMS Triangular Wave Time Function Block

Table 3. System Parameters

Component	Parameter	Value

Triangular Wave Time Function Block triang1	off	0 [S]
	tdelay	0 [S]
	freq	50 [Hz]
	ampl	326
	phase	0.0 [degree]

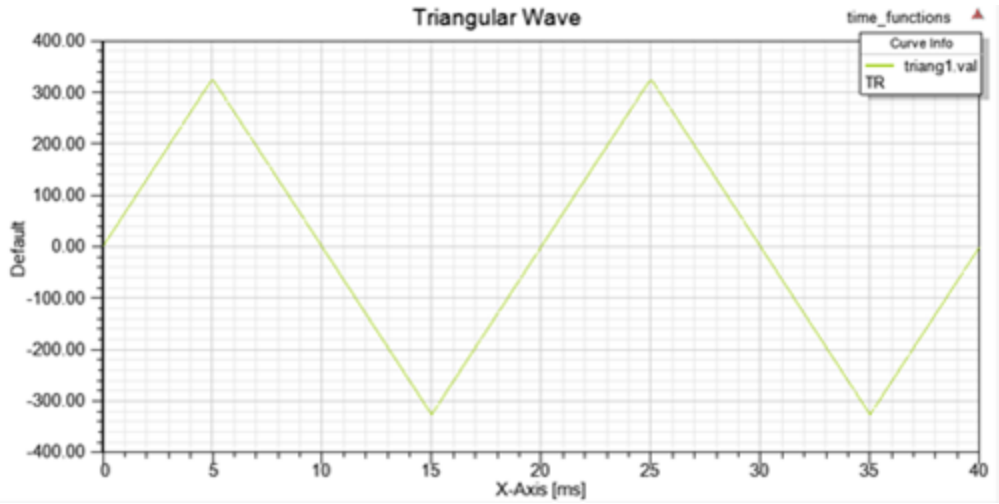


Figure 3. Simulation results-output from the Triangular Wave block.

Top

References

Index

A

Absolute Value Function 1-50

AMS Solenoid 1-382

Angle Source - VHDL-AMS 1-357

Angle Source/Spacer/Damper Rotational Displacement-Force Example 1-365

Angular Velocity Source - VHDL-AMS 1-363, 1-409

Angular Velocity Source/Spring/Damper Rotational Displacement-Force Example 1-367

Angular Velocity Source/Spring/Damper Rotational Velocity-Force Example 1-411

Arc Cosine Function 1-53

Arc Tangent 1-448

B

Basic Elements VHDLAMS Library 1-2

Bipolar Junction Transistor 1-195

C

Capacitor 1-176

Comparator 1-87

Conductor 1-181

Connecting Components - VHDL-AMS 1-7

Connecting Transfer Blocks 1-9

Constant Value 1-117

Continuous Blocks 1-12

Controlled Current Source 1-229

Controlled Voltage Source 1-217

Cosine Function 1-56

Current Controlled Switch 1-238

Current Source 1-225

D

Damper - VHDL-AMS 1-349, 1-372, 1-399, 1-416

DC Machine Linear Electrical Excitation 1-130

DC Machine Permanent Excitation 1-137

Defining the Sample Time 1-11

Delay 1-13

Derivative 1-16

Diode 1-200

Discrete Blocks 1-32

Discrete Derivative 1-33

Discrete Integrator 1-36

Displacement-Force Rotational Components 1-348

Displacement-Force Translational Components 1-371

Displacement-Force-Representation 1-347

E

Electrical Machines 1-129

Electrical Meters 1-300

Exponential Function 1-59

F

Flow Source - VHDL-AMS 1-325

Fluidic Components - VHDL-AMS 1-313

Fluidic Meters 1-302

Flux Source - VHDL-AMS 1-333

Force Source - VHDL-AMS 1-374, 1-418

Force Source/Mass/Limit Stop Translational Displacement-Force Example 1-395

Force Source/Mass/Limit Stop Translational Velocity-Force Example 1-430

G

Gain 1-19

H

Heat Flow Source - VHDL-AMS 1-436

Hydraulic Capacitance - VHDL-AMS 1-314

Hydraulic Inductance - VHDL-AMS 1-317

Hydraulic Resistance - VHDL-AMS 1-329

I

Ideal Switch 1-246

Ideal Transfer Switch 1-250

IGBT 1-204

Induction Machine with Squirrel Cage Rotor 1-144

Inductor 1-185

Integrator 1-25

L

Limit Stop - VHDL-AMS 1-353, 1-376, 1-403, 1-420

Limiter 1-90

Linear Two-Winding Transformer 1-264

Load Reference Arrow System of Circuit Components 1-127

Load Reference Arrow System of Physical Domains - VHDL-AMS 1-4

M

Magnetic Circuit Example - VHDL-AMS 1-343

Magnetic Components - VHDL-AMS 1-332

Magnetic Meters 1-304

Magneto-Motive Force Source - VHDL-AMS 1-336

Magnetoreluctance - VHDL-AMS 1-338

Mass - VHDL-AMS 1-355, 1-378, 1-405, 1-422

Math Blocks 1-49

Maximum of Input Signals 1-93

Mechanical Components - VHDL-AMS 1-346

Mechanical Meters 1-306

Memory 1-28

Minimum of Input Signals 1-97

Modeling Tools 1-445

Modeling with Block Diagrams 1-8

Modeling with Circuit Components 1-126

MOS Fieldeffect Transistor 1-208

Multiplier 1-101

N

Natural Logarithm Function 1-62

Nature Types of Components - VHDL-AMS 1-6

Needle 1-451

Negator 1-105

Network Configurations 1-128

P

Passive Elements 1-175

Position Source - VHDL-AMS 1-380

Position Source/Spacer/Damper Translational Displacement-Force Example 1-390

Power 1-77

Pressure Source - VHDL-AMS 1-321

Primary Side of a Six-Winding Transformer 1-273

Primary Side of a Two-Winding Transformer 1-258

Pulse 1-454

R

Random 1-120

Reciprocal Function 1-65

Resistor 1-190

Root 1-80

S

Sample and Hold Element 1-43

Saw-tooth Wave Falling 1-458

Saw-tooth Wave Rising 1-461

Secondary Side of a Six-Winding Transformer 1-278

Secondary Side of a Two-Winding Transformer 1-261

Semiconductors System Level 1-194

Sign 1-83

Signal Direction in Block Diagrams 1-10

Signal Processing Blocks 1-86

Sine Function 1-68

Sine Hyperbolic Function 1-71

Sine Wave 1-463

Single-Phase Systems 1-257

Single-Phase Transformer Example 1-269

Six-Winding Transformer Large-Power 1-283

Six-Winding Transformer Small-Power 1-290

Source Blocks 1-116

Sources 1-212

Spacer - VHDL-AMS 1-359, 1-384

Spring - VHDL-AMS 1-361, 1-386, 1-407, 1-424

Step Function 1-123

S-Transfer Function 1-22

Summation 1-108

Switches 1-237

Synchronous Machine Electrical Excitation with Damper 1-158

Synchronous Machine Electrical Excitation without Damper 1-151

Synchronous Machine Permanent Excitation with Damper 1-169

Synchronous Machine Permanent Excitation without Damper 1-164

T

Tangent Function 1-74

Temperature Source - VHDL-AMS 1-440

Thermal Capacitance - VHDL-AMS 1-433

Thermal Component Example 1-442

Thermal Components - VHDL-AMS 1-432

Thermal Meters 1-311

Thermal Resistance - VHDL-AMS 1-438

Three-Phase Six-Winding Transformer Example 1-297

Three-Phase Systems 1-272

Time Functions 1-447

Torque Source - VHDL-AMS 1-351, 1-401

Torque Source/Mass/Limit Stop Rotational Displacement-Force Example 1-369

Torque Source/Mass/Limit Stop Rotational Velocity-Force Example 1-413

Transformers 1-255

Trapezoidal Wave 1-466

Triangular Wave 1-469

Two-point Element with Hysteresis 1-112

U

Unit Delay 1-46

Using Measuring Instruments 1-299

Using Time Functions 1-446

V

Velocity Source - VHDL-AMS 1-388, 1-426

Velocity Source/Spring/Damper Translational Displacement-Force Example 1-393

Velocity Source/Spring/Damper Translational Velocity-Force Example 1-428

Velocity-Force-Representation 1-397

Voltage Controlled Oscillator Current Source 1-233

Voltage Controlled Oscillator
Voltage Source 1-221

Voltage Controlled Switch 1-242

Voltage Source 1-213

W

Winding - VHDL-AMS 1-340

Z

Z-Transfer Function (Discrete) 1-39