Twin Builder® Components: Digital Elements

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

1 - Digital Elements

The Digital Elements Library has been developed in VHDL to support users with the common basic functionality used for simple digital circuits. The models have been developed according to the IEEE 1076 (VHDL standard).

The digital components operate with digital signals and can be characterized with rise time/fall time/propagation delays. They do not have any conservative nodes but can be connected with analog quantities using omnicasters.

Digital VHDL components appear in the **Digital** tab. The components are subdivided into **ADC/DAC**, **Counters**, **Digital Sources**, **Flip-Flops**, **Latches**, **Logic Blocks**, and **Logic Gates**. Click the **Digital** folder, choose one of the groups, click on a name, drag the component onto the sheet, and release the mouse button.

The models provided are open source and can be used to derive more advanced models. This may be done by copying the element in the user library and editing the VHDL text to modify the model.

You can view the inputs and outputs of VHDL models using a 2D Digital Graph display element. You can also view the internal values used within the architecture of a model in a 2D Digital Graph and use them as variables between models. However, if the VHDL design needs to be exported to other simulators, then internal values of the model should not be referenced outside the model.

ADC and DAC Converters

Analog-Digital Converter

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

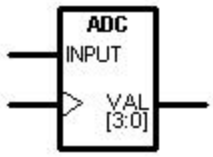

Figure 1. Component symbol

Description

The Analog-Digital Converter can be used in simple mixed signal circuit simulations. The component accepts an analog value input of type real and translates it to a four bit digital value according to specified minimum and maximum thresholds. A clock signal input is specified to sample the analog signal value.

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

COUPL adc ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (min := @min , max := @max , clk := @clk , 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]
min	Minimum Value of Analog Input	real	-1.0
max	Maximum Value of Analog Input	real	1.0
input	Analog Input Signal	real	0.0
clk	Clock	bit	'0'

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Digital Output Vector	Output	bit_vector

Top

Example

This example illustrates the use of the Analog-Digital Converter.

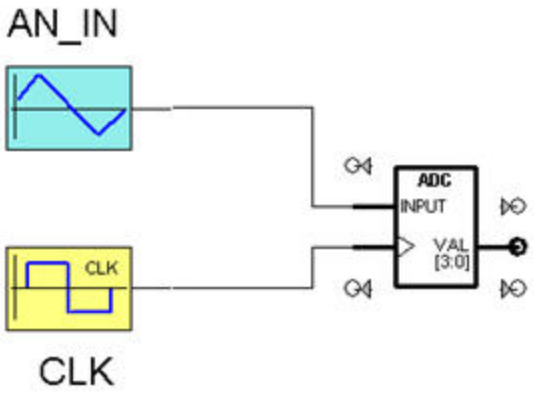

Figure 2. Application example of the Analog-Digital Converter

Table 3. System Parameters

Component	Parameter	Value [unit]
Clock Source CLK	ped	2m [S]
	del	0 [S]
	periodical	1.0
Triangular Wave Source AN_IN	ampl	1 [V}
	freq	10 [Hz]
	off	0 [V]
	tdelay	0 [S]
Analog-Digital Converter adc1	min	-1
	max	1
	clk	CLK.val
	input	AN_IN.val

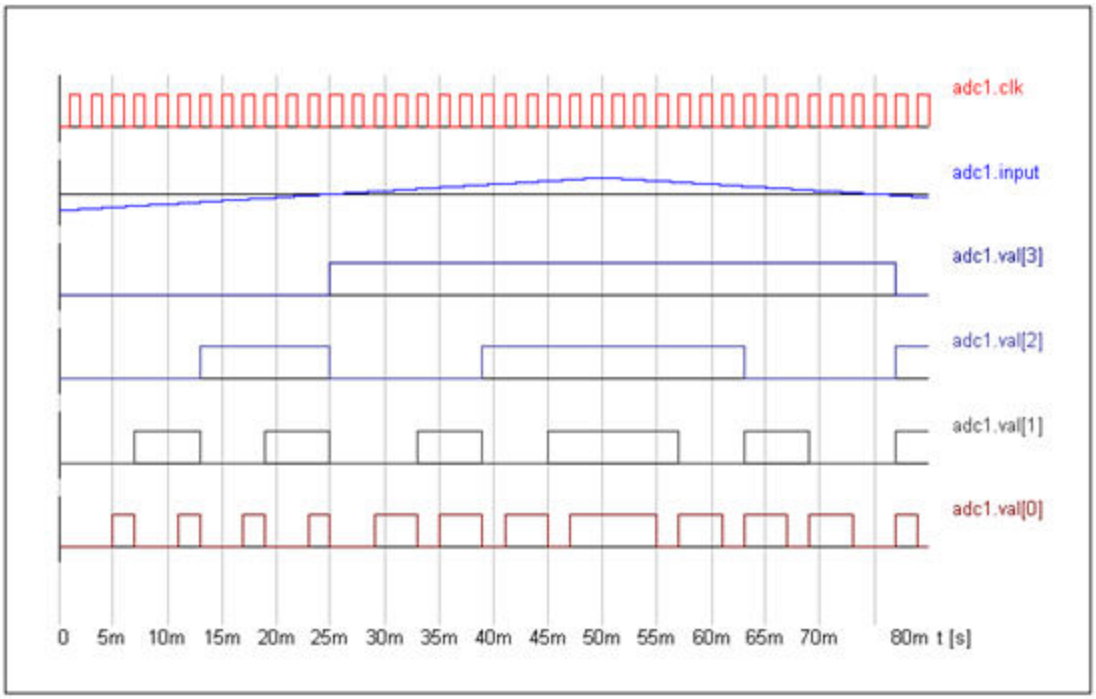

Figure 3. Simulation results-input and output of the Analog-Digital Converter

Top

References

adc8 : 8 Bit Analog-to-Digital Converter Model in VHDL-AMS

Library: Digital	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

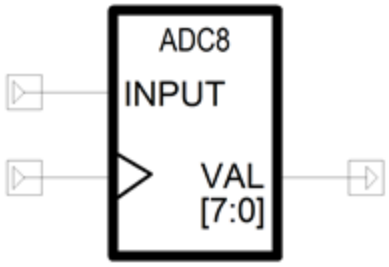

Figure 1. Component symbol

Description

The adc8 model can be used in simple mixed signal circuit simulations. The component accepts an analog value input of type real and translates it to an eight bit digital value according to specified minimum and maximum thresholds. A clock signal input is specified to sample the analog signal value.

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

Top

Parameters

Table 1

Name	Description	Data Type	Default Value [Unit]
min	Minimum Value of Analog Input	real	-1.0
max	Maximum Value of Analog Input	real	1.0
input	Analog Input Signal	real	0.0
clk	Clock	bit	'0'

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Digital Output Vector	Output	bit_vector

Top

Example

This example illustrates the use of the 8-bit Analog-Digital Converter.

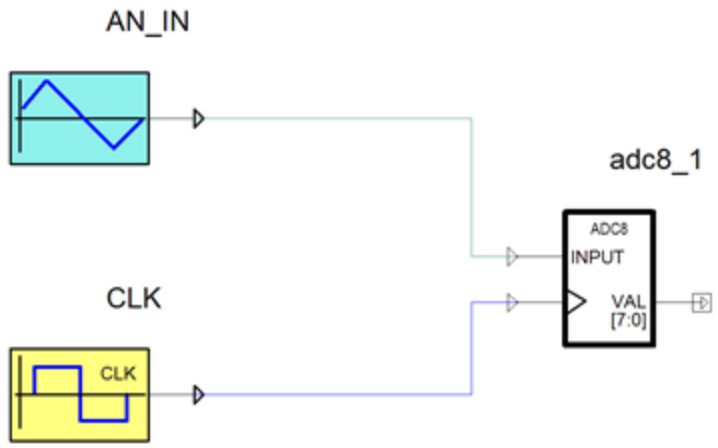

Figure 2. Application example of the 8-bit Analog-Digital Converter

Table 3. System Parameters

Component	Parameter	Default Value [unit]
Clock Source CLK	ped	0.0001

	del	0
	periodical	1
Triangular Wave Source AN_IN	ampl	1.0
	freq	10 [Hz]
	off	0
	tdelay	0
8-bit Analog-Digital Converter adc8	min	-1
	max	1
	input	AN_IN.val
	clk	CLK.val

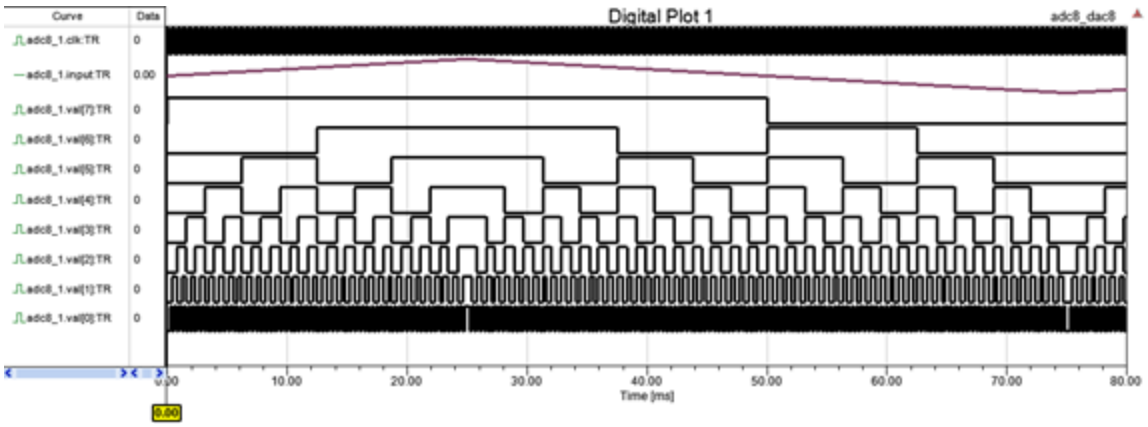

Figure 3. Simulation results-input and output of the 8-bit Analog-Digital Converter

Top

References

adc12: 12 Bit Analog-to-Digital Converter Model in VHDL-AMS

Library: Digital	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

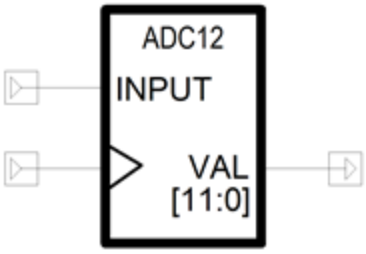

Figure 1. Component symbol

Description

The adc12 model can be used in simple mixed signal circuit simulations. The component accepts an analog value input of type real and translates it to a twelve-bit digital value according to specified minimum and maximum thresholds. A clock signal input is specified to sample the analog signal value.

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

Top

Parameters

Table 1

Name	Description	Data Type	Default Value [Unit]
min	Minimum Value of Analog Input	real	-1.0
max	Maximum Value of Analog Input	real	1.0
input	Analog Input Signal	real	0.0
clk	Clock	bit	'0'

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Digital Output Vector	Output	bit_vector

Top

Example

This example illustrates the use of the 12-bit Analog-Digital Converter.

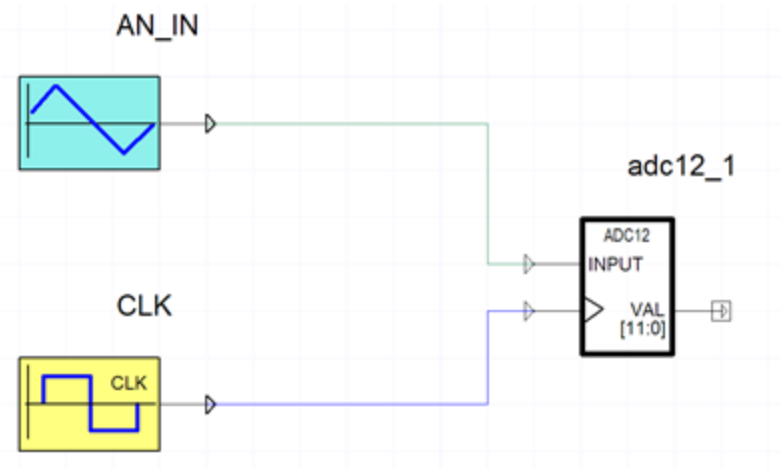

Figure 2. Application example of the 12-bit Analog-Digital Converter

Table 3. System Parameters

Component	Parameter	Default Value [unit]
Clock Source CLK	ped	1e-5

	del	0
	periodical	1
Triangular Wave Source AN_IN	ampl	1.0
	freq	10 [Hz]
	off	0
	tdelay	0
12-bit Analog-Digital Converter adc12	min	-1
	max	1
	input	AN_IN.val
	clk	CLK.val

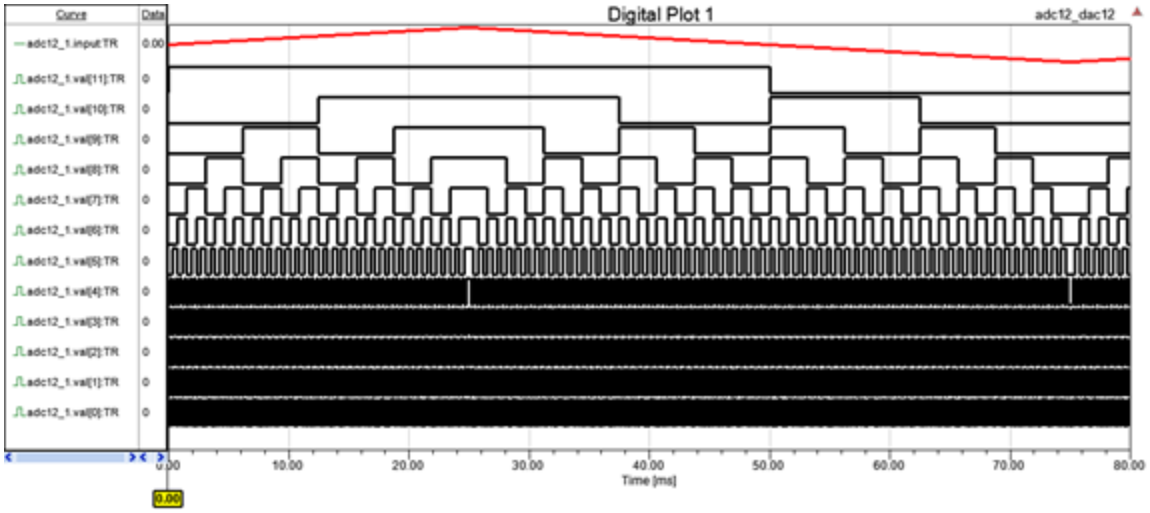

Figure 3. Simulation results-input and output of the 12-bit Analog-Digital Converter

Top

References

dac: Digital Analog Converter

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

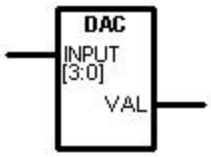

Figure 1. Component symbol

Description

The Digital-Analog Converter can be used in simple mixed signal circuit simulations. The component accepts a four bit digital input and translates it to an analog value of type real according to specified minimum and maximum thresholds.

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

COUPL dac ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (min := @min , max := @max , input := @input , input[3] := @input[3], input[2] := @input[2], input[1] := @input[1], input [0] := @input[0]) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:-:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value [Unit]
min	Minimum Value of Analog Output	real	-1.0
max	Maximum Value of Analog Output	real	1.0
input	4-Bit Digital Input Signal Vector	bit_vec-tor	'0'

Top

Input/Output Quantities

Table 2

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

Top

Example

This example illustrates the use of the Digital-Analog Converter.

Figure 2. Application example of the Digital-Analog Converter

Table 3. System Parameters

Component	Parameter	Value [unit]
Four-Bit Vector Source vec41	ped	3m [S]
	del	0 [S]
Digital-Analog Converter dac1	min	-1
	max	1

	input	vec41.val

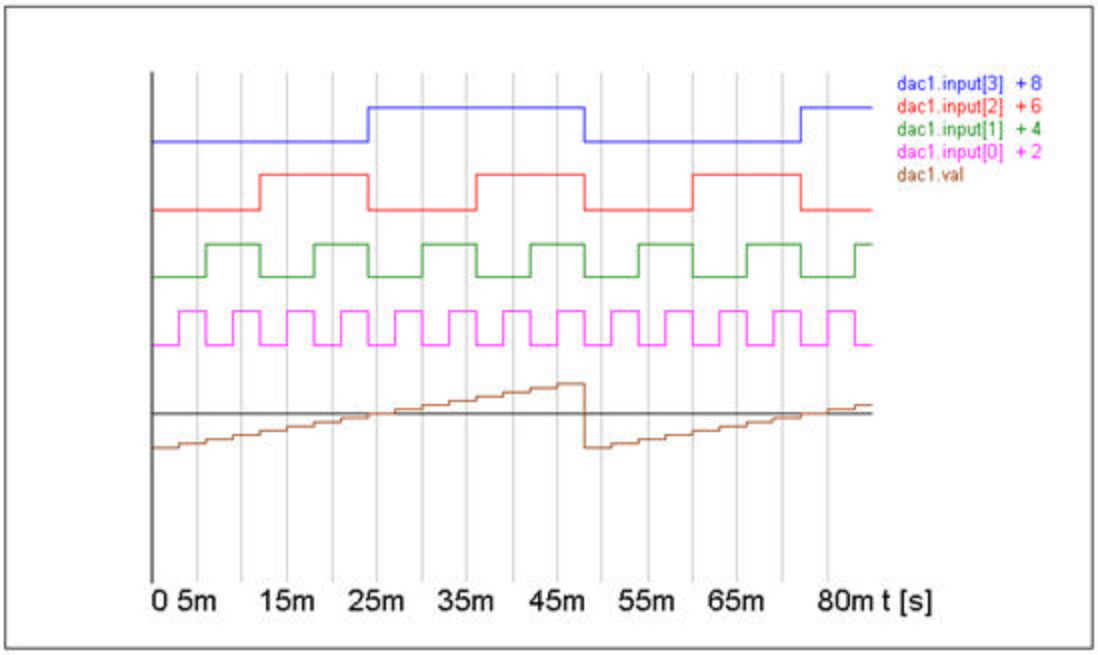

Figure 3. Simulation results-input and output of the Digital-Analog Converter

Top

References

dac8: 8 Bit Digital-to-Analog Converter Model in VHDL-AMS

Library: Digital	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

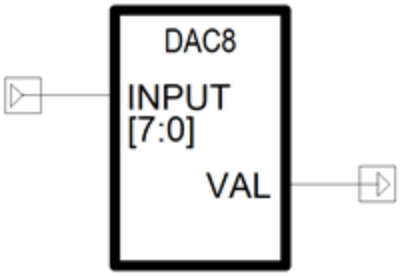

Figure 1. Component symbol

Description

The dac8 model can be used in simple mixed signal circuit simulations. The component accepts an 8-bit digital input and translates it to an analog value of type real according to specified minimum and maximum thresholds.

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

Top

Parameters

Table 1

Name	Description	Data Type	Default Value [Unit]
min	Minimum Value of Analog Output	real	-1.0
max	Maximum Value of Analog Output	real	1.0
input	8-bit Digital Input Signal Vector	bit_vector	'00000000'

Top

Input/Output Quantities

Table 2

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

Top

Example

This example illustrates the use of the 8-bit Digital-Analog Converter.

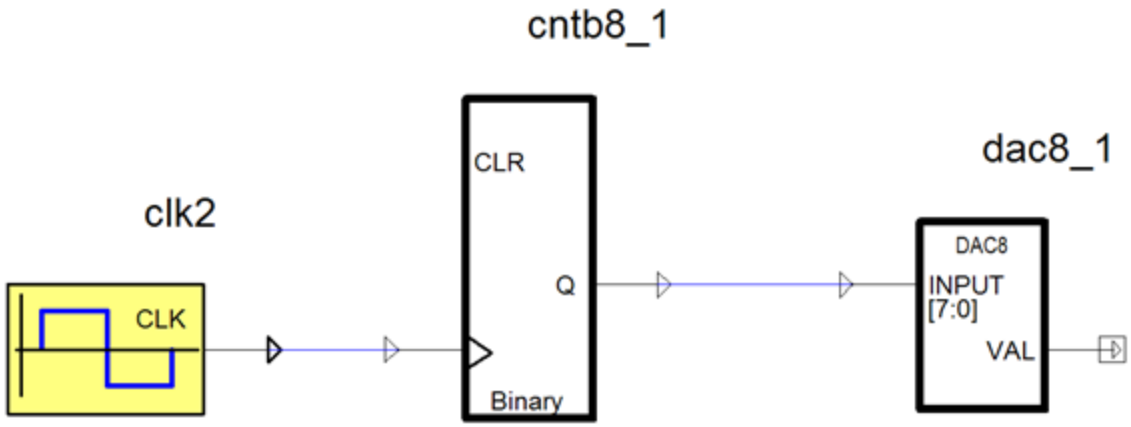

Figure 2. Application example of the 8-bit Digital-Analog Converter

Table 3. System Parameters

Component	Parameter	Value [unit]
Clock Source CLK	ped	0.0002

	del	0
	periodical	1
8-Bit Binary Counter cntb8	t_prop	0
	clk	clk2.val
	clr	0
8 Bit Analog-Digital Converter acd8	min	-1
	max	1
	input	cntb8_1.q

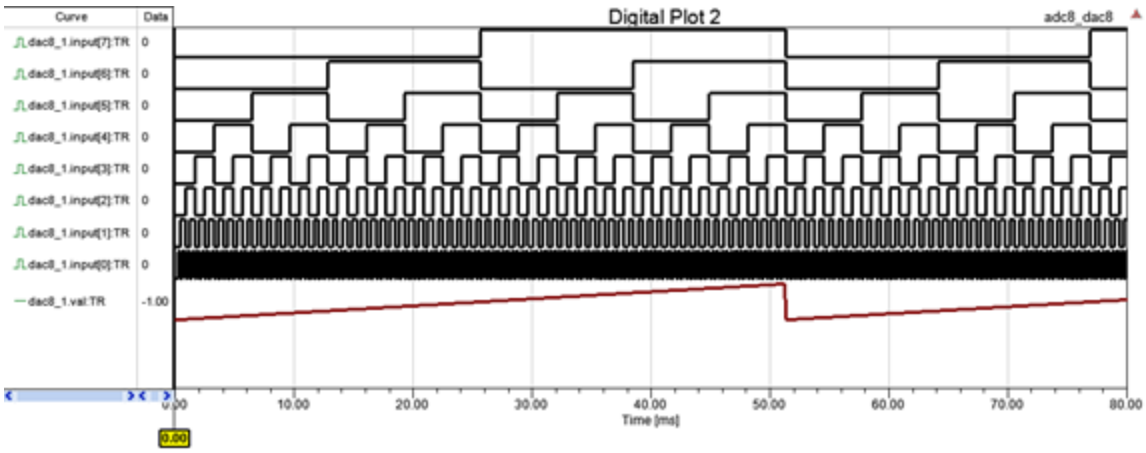

Figure 3. Simulation results-input and output of the 8-bit Digital-Analog Converter

Top

References

dac12: 12-bit Digital-to-Analog Converter Model in VHDL-AMS

Library: Digital	Modeling Language: VHDL-AMS	Version Number: Twin Builder 2025.2

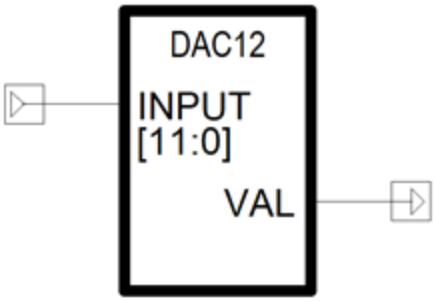

Figure 1. Component symbol

Description

The dac12 model can be used in simple mixed signal circuit simulations. The component accepts a 12-bit digital input and translates it to an analog value of type real according to specified minimum and maximum thresholds.

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

Top

Parameters

Table 1

Name	Description	Data Type	Default Value [Unit]
min	Minimum Value of Analog Output	real	-1.0
max	Maximum Value of Analog Output	real	1.0
input	12-bit Digital Input Signal Vector	bit_vector	'000000000000'

Top

Input/Output Quantities

Table 2

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

Top

Example

This example illustrates the use of the 12-bit Digital-Analog Converter.

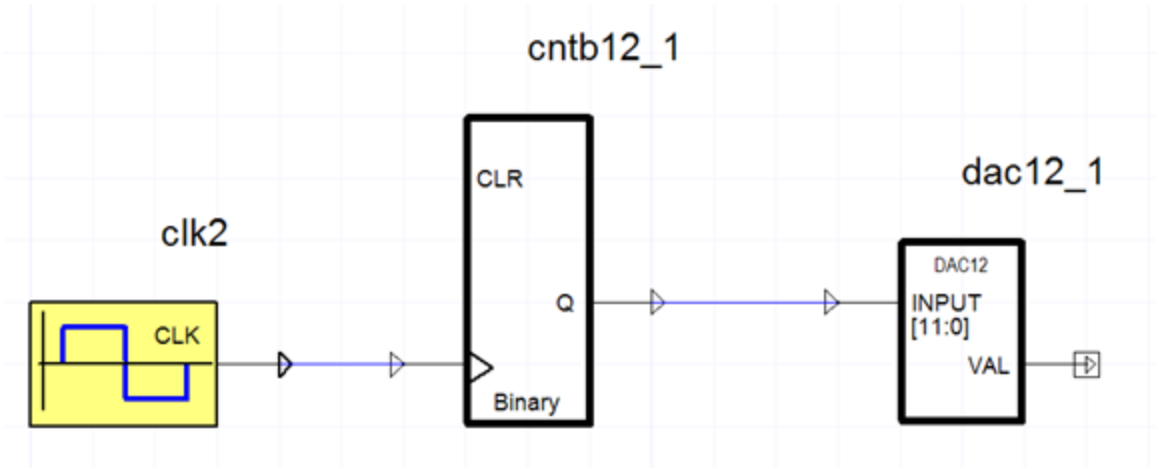

Figure 2. Application example of the 8-bit Digital-Analog Converter

Table 3. System Parameters

Component	Parameter	Value [unit]
Clock Source CLK	ped	1e-5

	del	0
	periodical	1
12-bit Binary Counter cntb12	t_prop	0
	clk	clk2.val
	clr	0
12-bit Analog-Digital Converteracd12	min	-1
	max	1
	input	cntb12_1.q

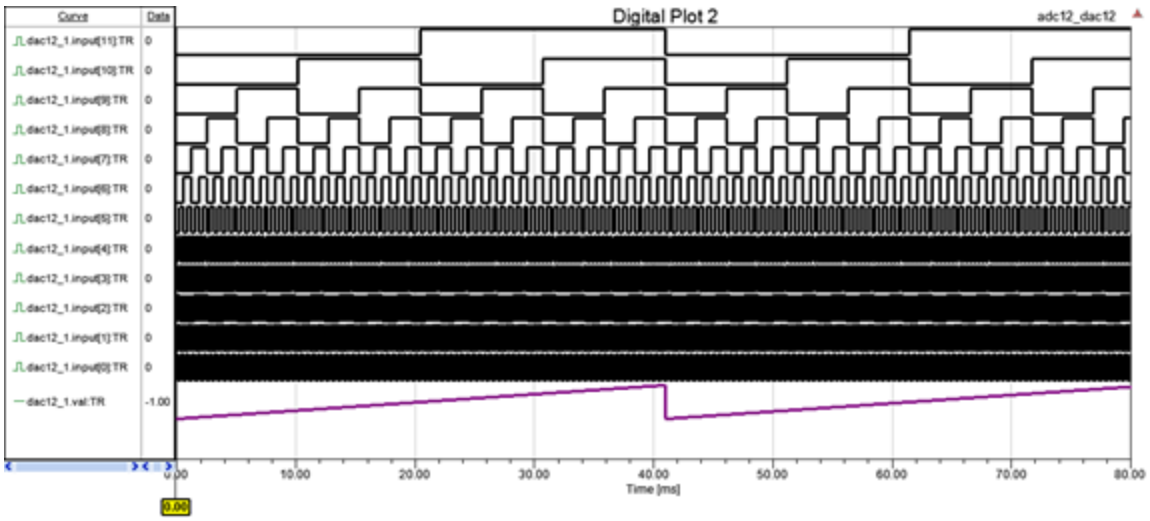

Figure 3. Simulation results-input and output of the 12-bit Digital-Analog Converter

Top

References

Counters

The counters can be used in simple sequential digital circuit simulations with propagation delays. The models operate on digital signals of type BIT and provide a vector output of the count value. Delays are specified in terms of propagation delays (**t_prop**).

The model is positive-edge triggered and has the capability of having the outputs cleared to LOW through an asynchronous **CLR** input.

- 2-bit binary counter with asynchronous clear input (cntb2)
- 3-bit binary counter with asynchronous clear input (cntb3)
- 4-bit binary counter with asynchronous clear input (cntb4)
- 8-bit BCD counter with asynchronous clear input (cntb8)
- 12-bit BCD counter with asynchronous clear input (cntb12)
- 4-bit BCD counter with asynchronous clear input (cntd4)

cntb12: 12-bit Binary Counter

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

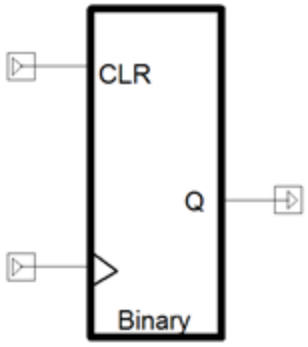

Figure 1. Component symbol

Description

The component represents a positive-edge triggered 12-bit binary counter that has the capability of having the outputs cleared to 000000000000 through an asynchronous CLR input.

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

Top

Parameters

Table 1

Name	Description	Data Type	Default Value [Unit]
clk	Clock	bit	'0'
clr	Clear	bit	'0'
t_prop	Propagation Delay	real	0.0 [sec]

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
q	Digital Output Vector	Output	bit_vector

Top

Example

This example illustrates the use of the 12-bit Binary Counter.

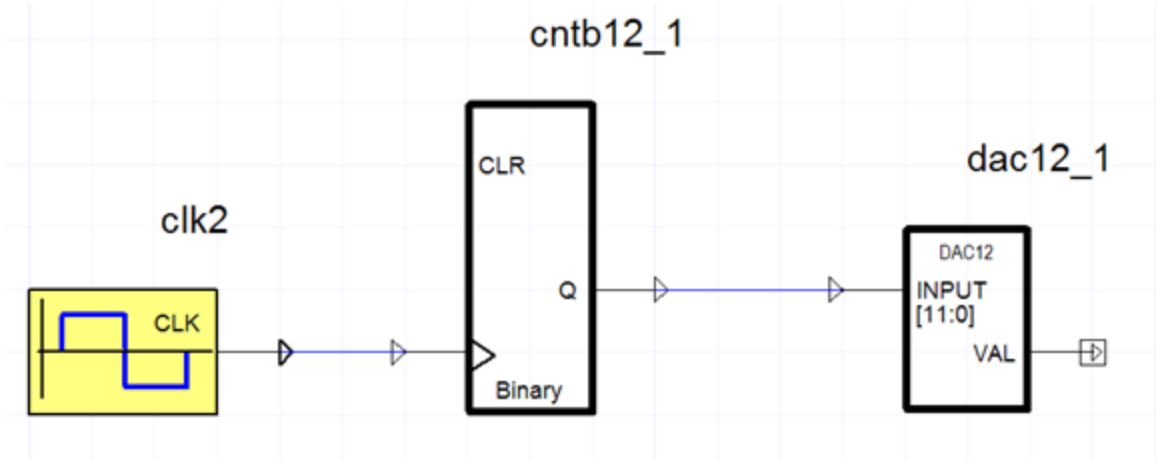

Figure 2. Application example of the 12-bit Binary Counter

Table 3. System Parameters

Component	Parameter	Value [unit]
Clock Source CLK	ped	1e-5
	del	0
	periodical	1
12-bit Binary Counter cntb12	t_prop	0
	clk	clk2.val
	clr	0
12-bit Digital-Analog Converter dac12	min	-1
	max	1
	input	cntb12_1.q

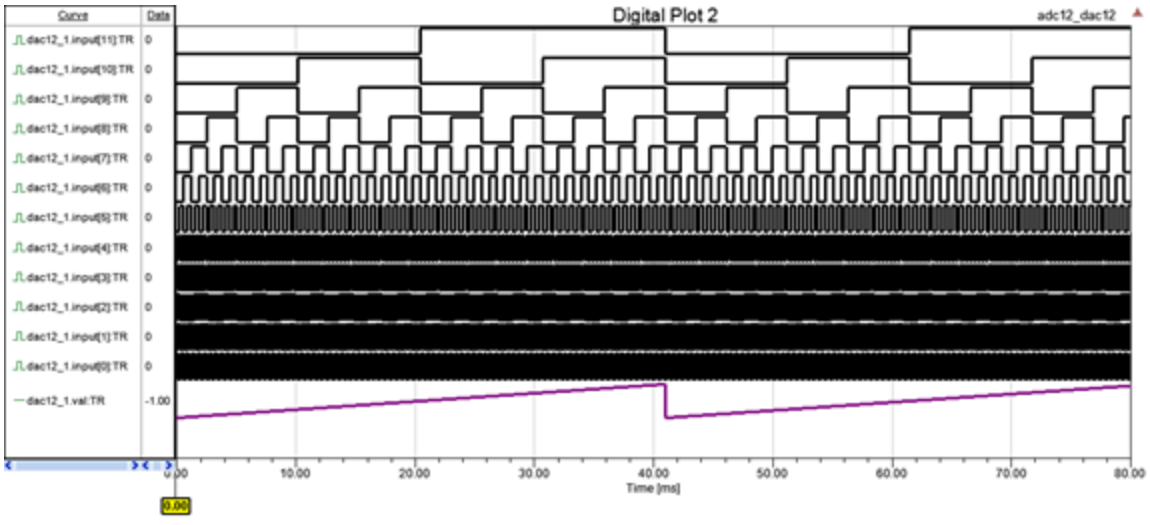

Figure 3. Simulation results-input and output of 12-bit Binary Counter

Top

References

cntb2: 2-bit Binary Counter

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

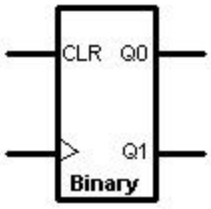

Figure 1. Component symbol

Description

The component represents a positive-edge triggered two-bit binary counter that has the capability of having the outputs cleared to 00 through an asynchronous CLR input.

Top

Assumptions and Limitations

Top

Mathematical Description

Table 1. Truth Table

Counting Sequence	
Q1	Q0
0	0
0	1
1	0
1	1
0	0

Top

Netlist Syntax

COUPL cntb2 ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (t_prop := @t_prop , clk := @clk , clr := @clr) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:-:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
clk	Clock	bit	'0'
clr	Clear	bit	'0'
t_prop	Propagation Delay	real	0.0 [s]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
q	Output Vector	Output	bit_vector

Top

Example

This example illustrates the use of the Two-Bit Binary Counter.

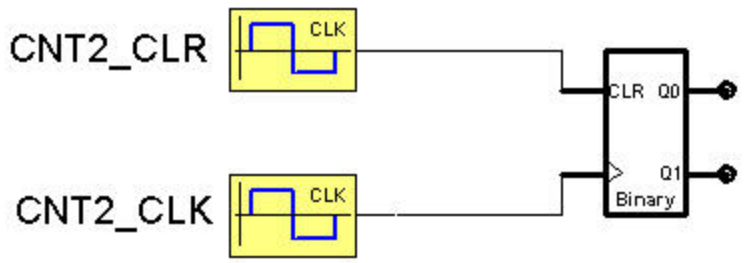

Figure 2. Application example of the Two-Bit Binary Counter

Table 4. System Parameters

Component	Parameter	Value [unit]
Clock Source CNT2_CLR	ped	10m [S]

	del	20m [S]
	periodical	0
Clock Source CNT2_CLK	ped	2m [S]
	del	0 [S]
	periodical	1.0
Two-Bit Binary Counter cntb21	t_prop	0.0 [S]
	clk	CNT2_CLK.val
	clr	CNT2_CLR.val

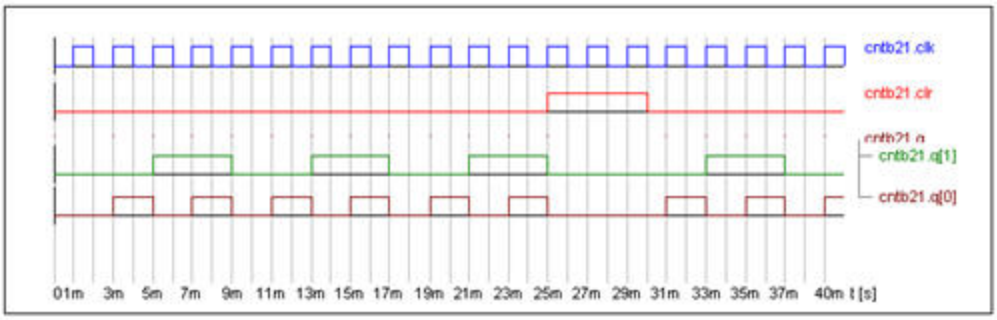

Figure 3. Simulation results-input and output of Two-Bit Binary Counter

Top

References

cntb3: Three-Bit Binary Counter

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

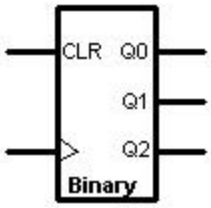

Figure 1. Component symbol

Description

The component represents a positive-edge triggered three-bit binary counter that has the capability of having the outputs cleared to 000 through an asynchronous **CLR** input.

Top

Assumptions and Limitations

Top

Mathematical Description

Table 1. Truth Table

Counting Sequence		
Q2	Q1	Q0
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0

1	0	1
1	1	0
1	1	1
0	0	0

Top

Netlist Syntax

COUPL cntb3 ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (t_prop := @t_prop , clk := @clk , clr := @clr) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:-:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
clk	Clock	bit	'0'
clr	Clear	bit	'0'
t_prop	Propagation Delay	real	0.0 [s]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
q	Output Vector	Output	bit_vector

Top

Example

This example illustrates the use of the Three-Bit Binary Counter.

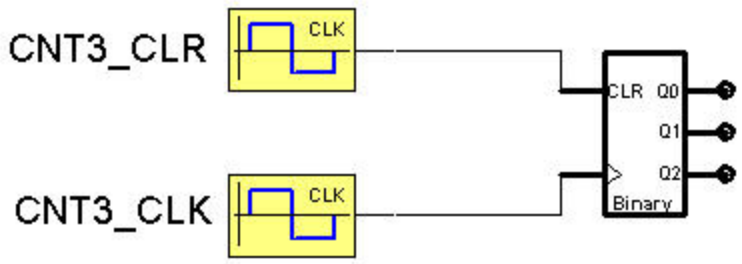

Figure 2. Application example of the Three-Bit Binary Counter

Table 4. System Parameters

Component	Parameter	Value [unit]
Clock Source CNT3_CLR	ped	10m [S]
	del	20m [S]
	periodical	0
Clock Source CNT3_CLK	ped	2m [S]
	del	0 [S]
	periodical	1.0
Three-Bit Binary Counter cntb31	t_prop	0.0 [S]
	clk	CNT3_CLK.val
	clr	CNT3_CLR.val

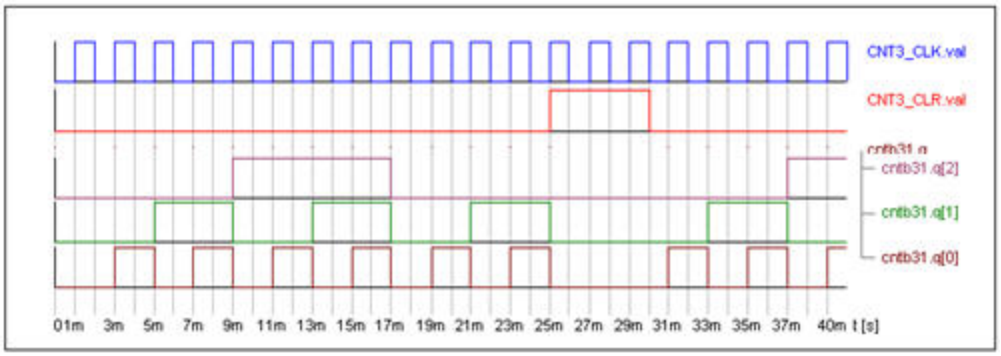

Figure 3. Simulation results-input and output of Three-Bit Binary Counter

Top

References

cntb4: Four-Bit Binary Counter

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

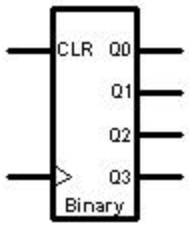

Figure 1. Component symbol

Description

The component represents a positive-edge triggered four-bit binary counter that has the capability of having the outputs cleared to 0000 through an asynchronous **CLR** input.

Top

Assumptions and Limitations

Top

Mathematical Description

Table 1. Truth Table

Counting Sequence			
Q3	Q2	Q1	Q0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0

0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1
0	0	0	0

Top

Netlist Syntax

COUPL cntb4 ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (t_prop := @t_prop , clk := @clk , clr := @clr) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:-:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
clk	Clock	bit	'0'
clr	Clear	bit	'0'
t_prop	Propagation Delay	real	0.0 [s]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
q	Output Vector	Output	bit_vector

Top

Example

This example illustrates the use of the Four-Bit Binary Counter.

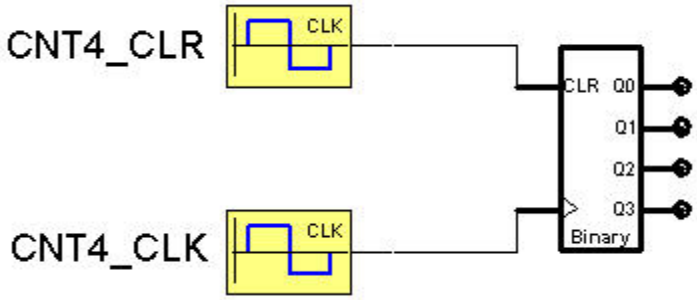

Figure 2. Application example of the Four-Bit Binary Counter

Table 4. System Parameters

Component	Parameter	Value [unit]
Clock Source CNT4_CLR	ped	10m [S]
	del	20m [S]
	periodical	0
Clock Source CNT4_CLK	ped	2m [S]
	del	0 [S]
	periodical	1.0
Four-Bit Binary Counter cntb41	t_prop	0.0 [S]
	clk	CNT4_CLK.val
	clr	CNT4_CLR.val

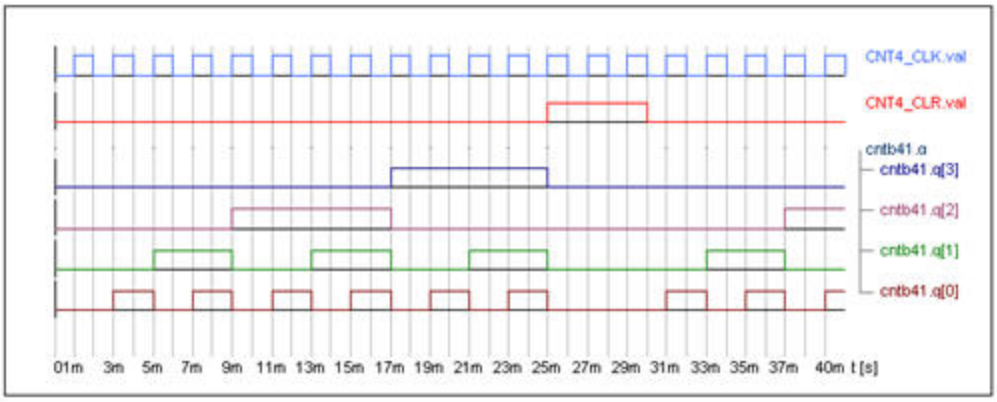

Figure 3. Simulation results-input and output of Four-Bit Binary Counter

Top

References

cntb8: 8-bit Binary Counter

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

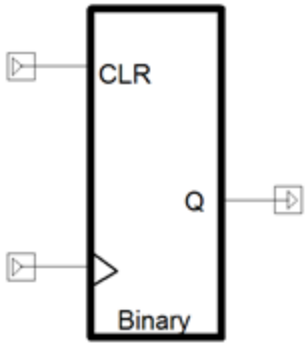

Figure 1. Component symbol

Description

The component represents a positive-edge triggered 8-bit binary counter that has the capability of having the outputs cleared to 00000000 through an asynchronous CLR input.

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

Top

Parameters

Table 1

Name	Description	Data Type	Default Value [Unit]
clk	Clock	bit	'0'
clr	Clear	bit	'0'
t_prop	Propagation Delay	real	0.0 [sec]

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
q	Digital Output Vector	Output	bit_vector

Top

Example

This example illustrates the use of the 8-bit Binary Counter.

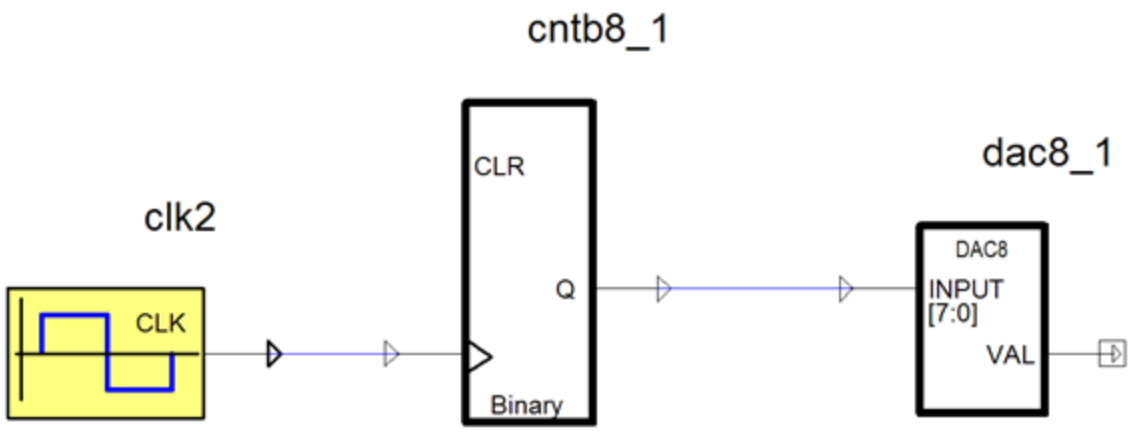

Figure 2. Application example of the 8-bit Binary Counter

Table 3. System Parameters

Component	Parameter	Value [unit]
Clock Source CLK	ped	0.0002
	del	0
	periodical	1

8-bit Binary Counter cntb8	t_prop	0
	clk	clk2.val
	clr	0
8-bit Digital-Analog Converter dac8	min	-1
	max	1
	input	cntb8_1.q

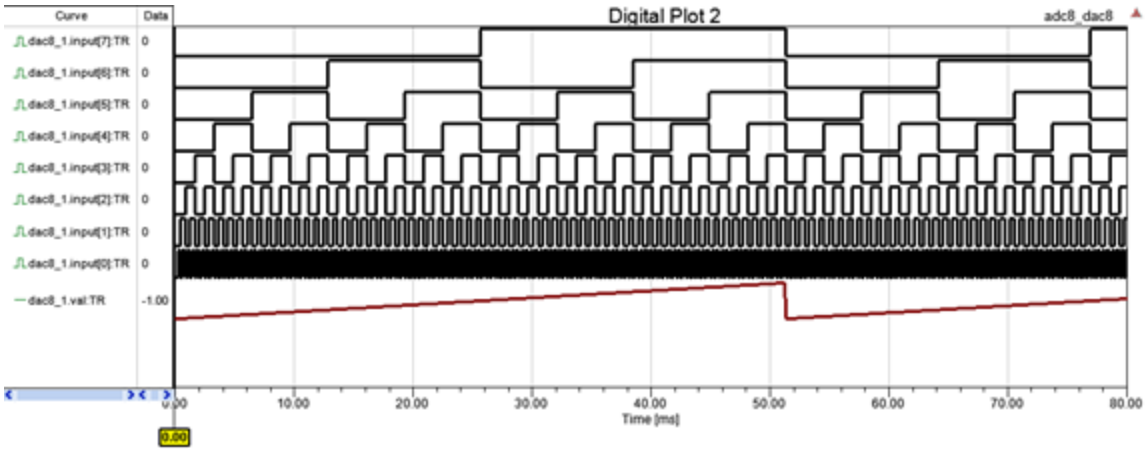

Figure 3. Simulation results-input and output of 8-bit Binary Counter

Top

References

cntd4: Four-Bit BCD Counter

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

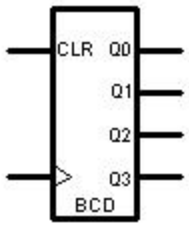

Figure 1. Component symbol

Description

The component represents a positive-edge triggered four bit BCD (divide by 10, binary coded decimal) counter that has the capability of having the outputs cleared to 0000 through an asynchronous **CLR** input.

Top

Assumptions and Limitations

Top

Mathematical Description

Table 1. Truth Table

Counting Sequence			
Q3	Q2	Q1	Q0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1

0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
0	0	0	0

Top

Netlist Syntax

COUPL cntd4 ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (t_prop := @t_prop , clk := @clk , clr := @clr) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:-:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
clk	Clock	bit	'0'
clr	Clear	bit	'0'
t_prop	Propagation Delay	real	0.0 [s]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
q	Output Vector	Output	bit_vector

Top

Example

This example illustrates the use of the Four-Bit BCD Counter.

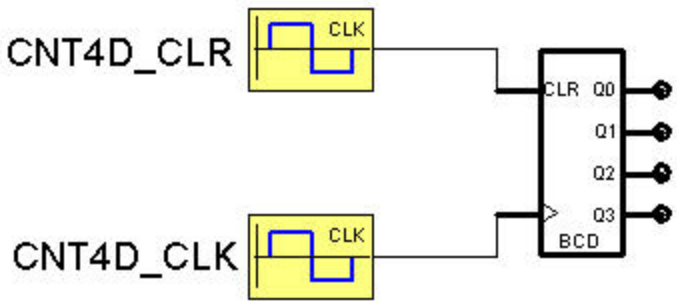

Figure 2. Application example of the Four-Bit BCD Counter

Table 4. System Parameters

Component	Parameter	Value [unit]
Clock Source CNT4D_CLR	ped	10m [S]
	del	20m [S]
	periodical	0
Clock Source CNT4D_CLK	ped	2m [S]
	del	0 [S]
	periodical	1.0
Four-Bit BCD Counter cntd41	t_prop	0.0 [S]
	clk	CNT4D_CLK.val
	clr	CNT4D_CLR.val

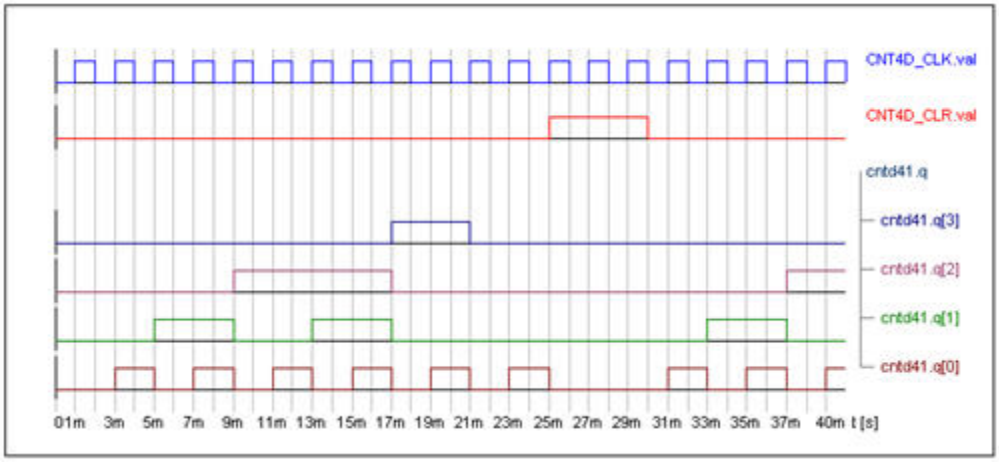

Figure 3. Simulation results-input and output of Four-Bit BCD Counter

Top

References

Digital Sources

The digital clock sources can provide input stimulus to digital models. The components can be parametrized with the clock period and the initial delay.

- Clock Source Model in VHDL (clk)
- Stimulus (Stimulus)
- Two-bit Vector Source Model in VHDL (vec2)
- Four-bit Vector Source Model in VHDL (vec4)

Clock Source

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

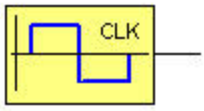

Figure 1. Component symbol

Description

The components represent a digital clock sources with an option of specifying whether the output signal is periodical. To define the values, enter a numerical value for time period (**ped**) and initial delay (**del**).

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

COUPL clk ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (ped := @ped , del := @del , periodical := @periodical) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value [Unit]
ped	Pulse Time Period	real	1.0e-3 [s]
del	Initial Delay	real	0.0 [s]
periodical	Periodical (1.0 -> yes, 0.0 -> no)	real	1.0

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Clock Output	Output	bit

Top

Example

See 2-to-4 Line Decoder Example

Top

References

Stimulus Source

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

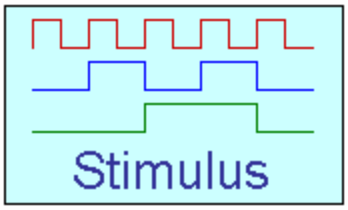

Figure 1. Component symbol

Description

The components represents a user defined arbitrary digital source with a user specified clocking pattern and output bit patterns. To define the base clock values, enter a time resolution, an end time, and the amount of time the stimulus should remain in a high and low position. Stimulus values are then defined in the Component Dialog.

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

Netlist generated by Special Component Dialog.

Top

Input/Output Quantities

Table 1

Name	Description [Unit]	Direction	Data Type
clk	Clock Output	Output	bit
bitsig *	Bit Signal Output	Output	bit
bvrsig *	Bit Vector Output	Output	Array of 4 bits
stdsig *	Std Logic Signal Output	Output	Std. Logic
svrsig *	Std Logic Vector Output	Output	Std. Logic

* Note: Only those signals defined by the user for each instance will be available.

Top

Example

Top

References

Component Dialog Settings

The following Parameters are set in the Component Dialog for the **Digital Stimulus Component**:

Name:

Specify the Name of the component instance. Select the **Show Name** checkbox to display the name on the schematic.

Parameters:

Base Clock: The base clock is represented in the dialog along with a time scale. To edit the **Base Clock** definition, click the button in the dialog to return to the Parameters Wizard. Create or delete entries of input signals with the symbols placed on the upper right side. Available signals are **Bit**, **Bit Vector**, **Std Logic**, or **Std Logic Vector**.

Base Clock Parameter Wizard: By default, you must first specify the Base Clock cycle for the Stimulus source. Note that the number of clock cycles that can be displayed is limited to 1000. A message is displayed informing you when the limiting is applied.

Time Resolution: Select the time resolution units for the pull-down list.

Stimulus End Time: Enter a numerical value (non-negative integer) in the text box. Please note: You cannot define a variable or expression in this case. Only a positive integer is accepted.

Clock High Time:/Clock Low Time: Type a numerical value (non-negative integer) in the text box to define the portion of time to remain at the high/low output value. Please note: You cannot define a variable or expression in this case. Only a number is accepted.

Start with Low Time: Select the checkbox to have the simulator start the stimulus source with the low time portion of the cycle.

BIT/Std Logic signal: Select the Wizard checkbox to enable the Parameters Wizard for the **BIT** source type or **Std Logic** source type. Leaving the Wizard checkbox unselected allows you to change the data for the signal manually for each time step. The **Bit** signal and the **Std Logic** are equivalent with the exception of allowable states. The **BIT** source can accept a logic "0" or logic "1" state only. The **Std Logic** source can have "U", "X", "0", "1", "Z", "W", "L", "H", and "-" states.

BIT/Std Logic Signal Parameter Wizard:

Signal Name: Enter a name for the signal.

Pattern Type: Three pattern types are supported for the bit signal, **Toggle**, **Pulse**, and **Random**. The selected pattern type and representation of the Properties is shown in the preview

window of the dialog.

The **Toggle** type switches between the two states on the clock signal trailing edge. The frequency of the switch is set by the Toggle Period parameter.
The **Pulse** type switches between two states with a user definable initial delay and pulse width.
The **Random** signal type uses a random number generator to select either a high or low value for the next random period specified by the user.

Pattern Repetition Number: Enter the number of times the pattern should repeat.

Initial Value: The value the signal takes at the beginning of the simulation. This is set to a logic "0" or "1".

Other Value/Pulse Value: The value the signal will switch to during clocking. This is set to a logic "0" or "1" and should be set to the opposite of the Initial Value setting.

Toggle Period/Pulse Width: The Toggle Period/Pulse Width defines how many clock cycles the pulse remains "ON" for the Toggle/Pulse pattern types respectively.

Initial Delay: In Pulse mode, this parameter defines the delay from the end of the last pulse, to the leading edge of the next pulse in clock cycles.

Random Seed: The random seed is a parameter used to seed a random number generator and is used to randomize the switching pattern of the signal. The random number generator returns only one number and all random signals in all stimulus components in the circuit will get the same value when determining whether to switch states.

Random Period: The random period is the length of the pulse before another switch is possible. Switching may or may not occur due to the random nature of the signal.

BIT/Std Logic Vector: The Bit Vector source is set up via the Bit Vector Parameter Wizard and cannot be manually modified.

BIT/Std Logic Vector Parameter Wizard:

Signal Name: Enter a name for the signal.

Pattern Type: Six pattern types are supported for the bit vector signal, **Alternate**, **Count Up**, **Count Down**, **Shift Left**, **Shift Right**, and **Random**. The selected pattern type and representation of the Properties is shown in the preview window of the dialog.

The **Alternate** type switches between two states on the clock signal trailing edge. The frequency of the switch is set by the toggle Period parameter.
The **Count Up/Count Down** performs a binary count between the initial and final values specified.
The **Shift Left/Shift Right** performs a standard binary shift operation, shifting the bits set in the initial value left or right and shifting in the value specified in the **Input Value**.
The **Random** signal type uses a random number generator to select either a high or low value

for the next random period specified by the user.

Pattern Repetition Number: Enter the number of times the pattern should repeat.

Vector Length: Enter the number of signals in the vector.

Value1/Pulse Width Value 1: For each **Index**, corresponding to the **Vector Length**, select the quantity for Value 1 and the associated **Pulse Width** as shown in the preview window.

Value2/Pulse Width Value 2: For each **Index**, corresponding to the **Vector Length**, select the quantity for Value 2 and the associated **Pulse Width** as shown in the preview window. Generally, Value 2 should be set to the opposite of Value 1 for each index entry.

Initial Value: When the **Pattern Type** is either **Count Up, Count Down, Shift Left,** or **Shift Right**, specify the bit values for the initial vector.

Final Value: When the **Pattern Type** is either **Count Up** or **Count Down**, specify the bit values for the final vector.

Increment/Decrement Value: Specify the increment for each count in the **Count Up** or **Count Down** sequence. An **Increment Value** of 1 will provide a standard binary count sequence.

Count: Specify the number of clock cycles per count. A **Count** of one will increment or decrement the vector with each clock cycle.

Input Value: When using the **Shift Left** or **Shift Right** type, the **Input Value** indicates what data value is to be shifted into the vector.

Random Seed: The random seed is a parameter used to seed a random number generator and is used to randomize the switching pattern of the signal. The random number generator returns only one number and all random signals in all stimulus components in the circuit will get the same value when determining whether to switch states.

Random Period: The random period is the length of the pulse before another switch is possible. Switching may or may not occur due to the random nature of the signal.

Random Value: The random value allows you to choose between randomly setting the vector with "0", "1" states only, or using all possible logic states.

Outputs/Display

For information regarding displaying parameter pins, see *Output/Display Tab* in the main Twin Builder help.

Two-bit Vector Source

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

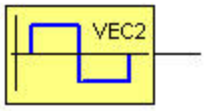

Figure 1. Component symbol

Description

The component represents a two-bit vector source that can provide input stimulus of 00,01,10,11 at specified time intervals. To define the values, enter a numerical value for the period of each vector value (**ped**) and initial delay (**del**).

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

COUPL vec2 ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (ped := @ped , del := @del) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:-:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value [Unit]
ped	Pulse Time Period	real	1.0e-3 [s]
del	Initial Delay	real	0.0 [s]

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Two-bit Vector Output	Output	Bit_vector

Top

Example

See Two-Bit Comparator Example

Top

References

Four-bit Vector Source

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

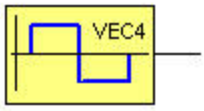

Figure 1. Component symbol

Description

The component represents a four-bit vector source that can provide input stimulus from 0000 to 1111 at specified time intervals. To define the values, enter a numerical value for the period of each vector value (**ped**) and initial delay (**del**).

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

COUPL vec4 ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (ped := @ped , del := @del) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:-:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value [Unit]
ped	Pulse Time Period	real	1.0e-3 [s]
del	Initial Delay	real	0.0 [s]

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
val	Four-bit Vector Output	Output	Bit_vector

Top

Example

See Four Bit Bidirectional Serial Shift Register Example

Top

References

Flip Flops

The flip-flop models can be used in simple sequential digital circuit simulations with propagation delays. The models operate on digital signals of type **BIT**. Delays are specified in terms of propagation delays (**t_prop**).

All models are positive edge triggered and some have the capability of having the outputs preset to '1' or cleared to '0' through asynchronous **PST** and **CLR** inputs. If both **PST** and **CLR** inputs are active at the same time, priority is given to the **PST** input. The following three types of flip-flop models are provided:

- D Flip-Flop
- JK Flip-Flop
- T Flip-Flop

D-Flip Flops

Basic D-Flip-Flop

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

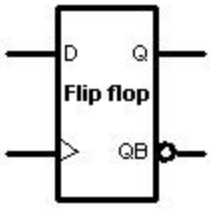

Figure 1. Component symbol

Description

The component represents a positive-edge triggered basic D Flip-Flop.

Top

Assumptions and Limitations

Top

Mathematical Description

Table 1. Truth Table

CLK	D	Q	QB
0	D	NC	NC
rising	0	0	1
rising	1	1	0
NC=No Change			

Top

Netlist Syntax

COUPL ffd ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (t_prop := @t_prop , clk := @clk , din := @din) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:-:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
clk	Clock input, data transferred at positive edge	bit	'0'
din	Data Input	bit	'0'
t_prop	Propagation Time	real	0.0 [s]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
q	Data Output	Output	bit
qb	Complemented Data Output	Output	bit

Top

Example

This example illustrates the use of the Basic D-Flip-Flop.

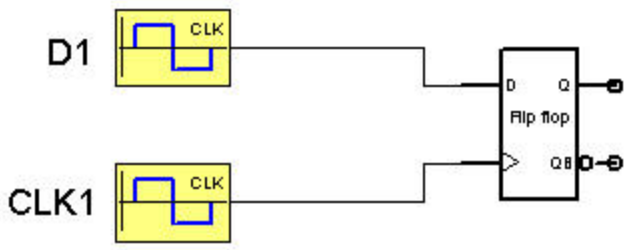

Figure 2. Application example of the Basic D-FlipFlop

Table 4. System Parameters

Component	Parameter	Value [unit]
Clock Source D1		

	ped	12m [S]
	del	0 [S]
	periodical	1.0
Clock Source CLK1	ped	8m [S]
	del	0 [S]
	periodical	1.0
Basic D-Flip-Flop ffd1	t_prop	0.0 [S]
	clk	CLK1.val
	din	D1.val

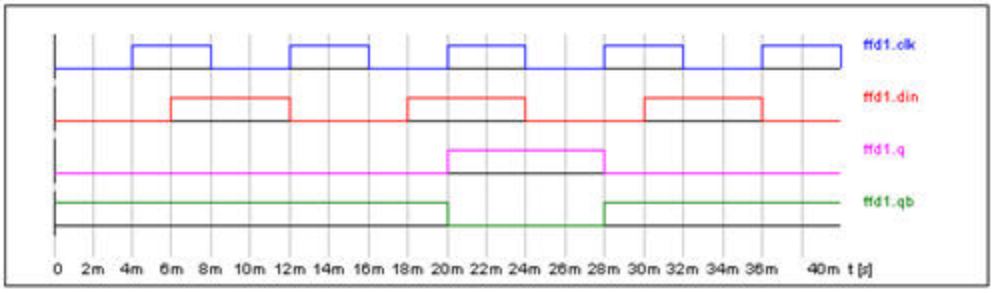

Figure 3. Simulation results-input and output of Basic D-Flip-Flop

Top

References

Basic D-Flip-Flop with Preset and Clear

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

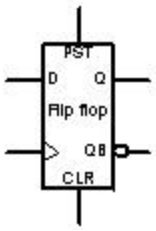

Figure 1. Component symbol

Description

The component represents a positive-edge triggered basic D Flip-Flop with preset and clear.

Top

Assumptions and Limitations

Top

Mathematical Description

Table 1. Truth Table

CLK	CLR	PST	D	Q	QB
0	0	0	D	NC	NC
X	1	0	D	0	1
X	0	1	D	1	0
rising	0	0	0	0	1
rising	0	0	1	1	0
NC=No Change					
X=Don’t Care					

Top

Netlist Syntax

COUPL ffdcp ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (t_prop := @t_prop , clk := @clk , din := @din , clr := @clr , pst := @pst) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
clk	Clock input, data transferred at positive edge	bit	'0'
din	Data Input	bit	'0'
pst	Asynchronous Active-High Pre-set Signal	bit	'0'
clr	Asynchronous Active-High Clear Signal	bit	'0'
t_prop	Propagation Time	real	0.0 [s]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
q	Data Output	Output	bit
qb	Complemented Data Output	Output	bit

Top

Example

This example illustrates the use of the Basic D-Flip-Flop with Preset and Clear.

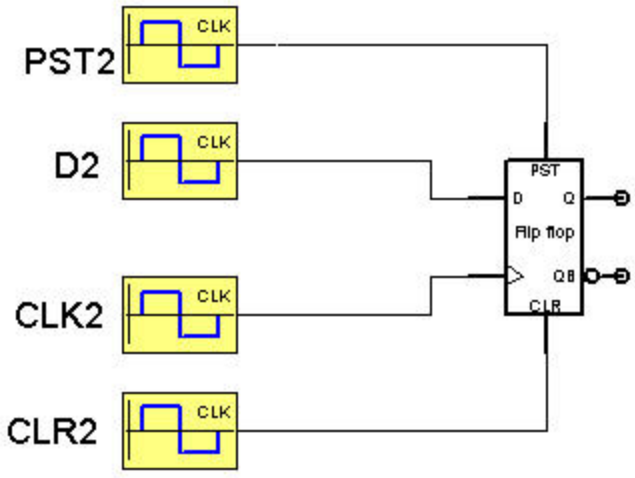

Figure 2. Application example of the Basic D-Flip-Flop with Preset and Clear

Table 4. System Parameters

Component	Parameter	Value [unit]
Clock Source D2	ped	10m [S]
	del	0 [S]
	periodical	1.0
Clock Source CLK2	ped	4m [S]
	del	0 [S]
	periodical	1.0
Clock Source CLR2	ped	10m [S]
	del	15m [S]
	periodical	0
Clock Source PST2	ped	8m [S]
	del	20m [S]

	periodical	0
D-Flip-Flop with Preset and Clear ffdcp1	t_prop	0.0 [S]
	clk	CLK2.val
	din	D2.val
	clr	CLR2.val
	pst	PST2.val

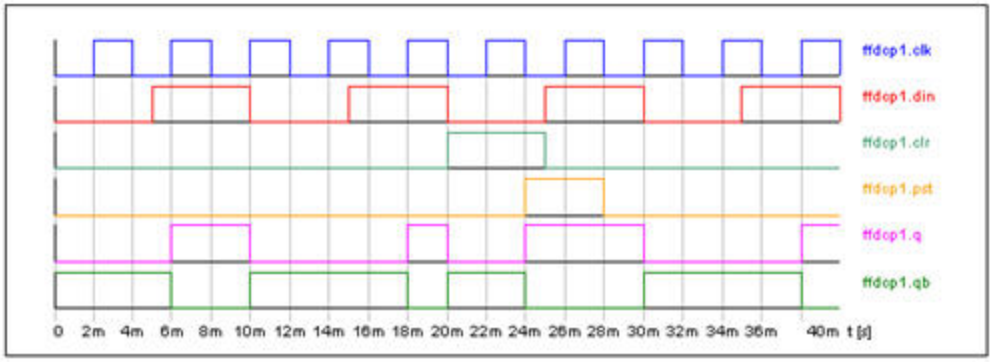

Figure 3. Simulation results-input and output of Basic D-Flip-Flop with Preset and Clear

Top

References

D-Flip-Flop with Active-Low Preset and Clear

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

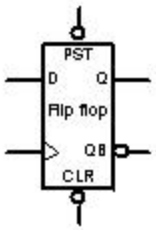

Figure 1. Component symbol

Description

The component represents a positive-edge triggered basic D Flip-Flop with active-low preset and clear.

Top

Assumptions and Limitations

Top

Mathematical Description

Table 1. Truth Table

CLK	CLR	PST	D	Q	QB
0	1	1	D	NC	NC
X	0	1	D	0	1
X	1	0	D	1	0
rising	1	1	0	0	1
rising	1	1	1	1	0
NC=No Change					
X=Don’t Care					

Top

Netlist Syntax

COUPL ffdcpal ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (t_prop := @t_prop , clk := @clk , din := @din , clr := @clr , pst := @pst) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
clk	Clock input, data transferred at positive edge	bit	'0'
din	Data Input	bit	'0'
pst	Asynchronous Active-Low Pre-set Signal	bit	'1'
clr	Asynchronous Active-Low Clear Signal	bit	'1'
t_prop	Propagation Time	real	0.0 [s]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
q	Data Output	Output	bit
qb	Complemented Data Output	Output	bit

Top

Example

This example illustrates the use of the D-Flip-Flop with Active-Low Preset and Clear.

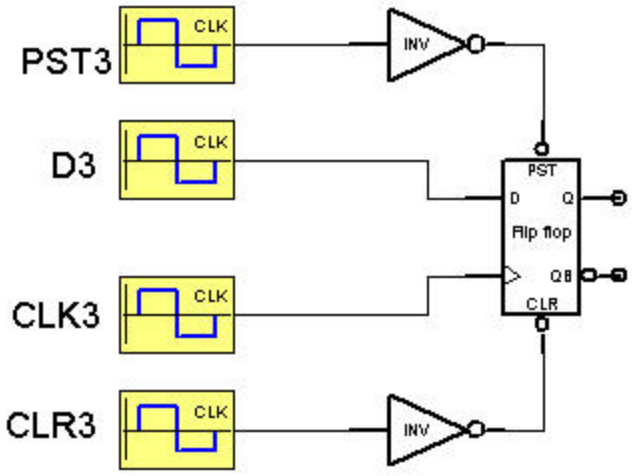

Figure 2. Application example of the D-Flip-Flop with Active Low Preset and Clear

Table 4. System Parameters

Component	Parameter	Value [unit]
Clock Source D3	ped	10m [S]
	del	0 [S]
	periodical	1.0
Clock Source CLK3	ped	4m [S]
	del	0 [S]
	periodical	1.0
Clock Source CLR3	ped	10m [S]
	del	15m [S]
	periodical	0
Clock Source PST3	ped	8m [S]
	del	20m [S]
	periodical	0
Inverter inv1	tp_lh	0 [S]
	tp_hl	0 [S]
	x	CLR3.val
Inverter inv2	tp_lh	0 [S]
	tp_hl	0 [S]
	x	PST3.val

D-Flip-Flop with Active-Low Preset and Clear ffdcpal1	t_prop	0.0 [S]
	clk	CLK3.val
	din	D3.val
	clr	inv1.y
	pst	inv2.y

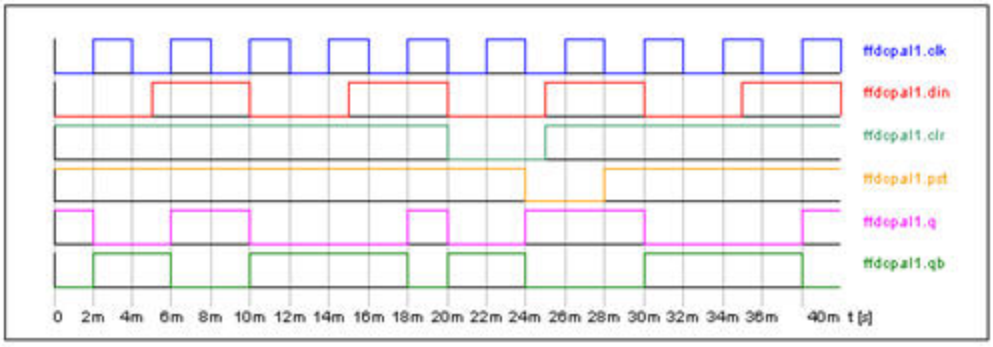

Figure 3. Simulation results-input and output of D-Flip-Flop with Active-Low Preset and Clear

Top

References

JK-Flip Flops

Basic JK-Flip-Flop

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

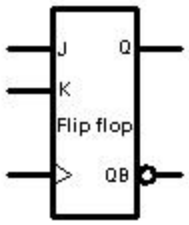

Figure 1. Component symbol

Description

The component represents a positive-edge triggered basic JK Flip-Flop.

Top

Assumptions and Limitations

Top

Mathematical Description

Table 1. Truth Table

CLK	J	K	Q	QB
0	J	K	NC	NC
rising	0	0	NC	NC
rising	0	1	0	1
rising	1	0	1	0
rising	1	1	Q'	QB'
NC=No Change				

Top

Netlist Syntax

COUPL ffjk ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (t_prop := @t_prop , clk := @clk , j := @j , k := @k) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
clk	Clock input, data transferred at positive edge	bit	'0'
jin	J Input	bit	'0'
kin	K Input	bit	'0'
t_prop	Propagation Time	real	0.0 [s]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
q	Data Output	Output	bit
qb	Complemented Data Output	Output	bit

Top

Example

This example illustrates the use of the JK-Flip-Flop.

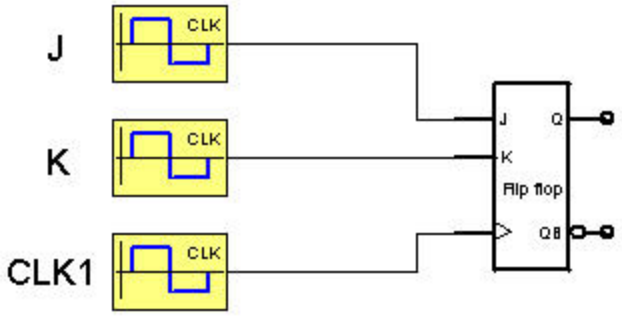

Figure 2. Application example of the JK-Flip-Flop

Table 4. System Parameters

Component	Parameter	Value [unit]
Clock Source J	ped	8m [S]
	del	0 [S]
	periodical	1.0
Clock Source CLK1	ped	4m [S]
	del	0 [S]
	periodical	1.0
Clock Source K	ped	16m [S]
	del	0 [S]
	periodical	1.0
JK-Flip-Flop ffjk1	t_prop	0.0 [S]
	clk	CLK1.val
	j	J.val
	k	K.val

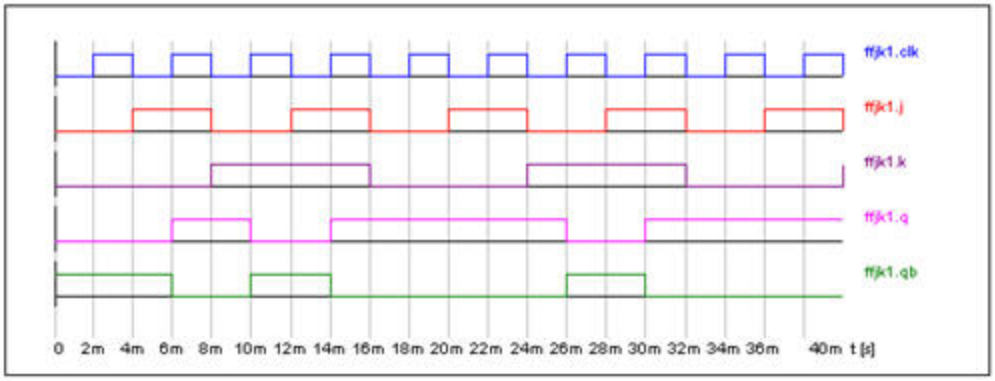

Figure 3. Simulation results-input and output of JK-Flip-Flop

Top

References

JK-Flip-Flop with Preset and Clear

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

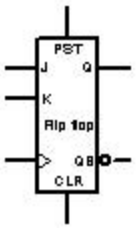

Figure 1. Component symbol

Description

The component represents a positive-edge triggered basic JK Flip-Flop with preset and clear.

Top

Assumptions and Limitations

Top

Mathematical Description

Table 1. Truth Table

CLK	CLR	PST	J	K	Q	QB
0	0	0	J	K	NC	NC
X	1	0	J	K	0	1
X	0	1	J	K	1	0
rising	0	0	0	0	NC	NC
rising	0	0	0	1	0	1
rising	0	0	1	0	1	0
rising	0	0	1	1	Q'	QB'

NC=No Change	
X=Don’t Care	

Top

Netlist Syntax

COUPL ffjkcp ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (t_prop := @t_prop , clk := @clk , j := @j , k := @k , clr := @clr , pst := @pst) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
clk	Clock input, data transferred at positive edge	bit	'0'
jin	J Input		'0'
kin	K Input	bit	'0'
pst	Asynchronous Active-High Pre-set Signal	bit	'0'
clr	Asynchronous Active-High Clear Signal	bit	'0'
t_prop	Propagation Time	real	0.0 [s]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
q	Data Output	Output	bit
qb	Complemented Data Output	Output	bit

Top

Example

This example illustrates the use of the JK-Flip-Flop with Preset and Clear.

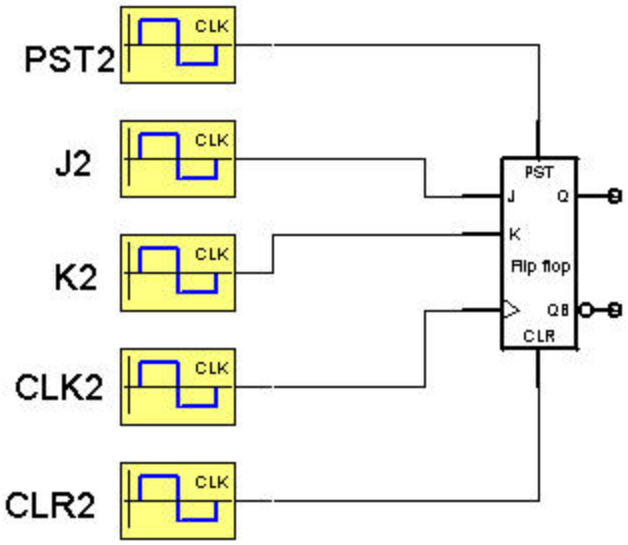

Figure 2. Application example of the JK-Flip-Flop with Preset and Clear

Table 4. System Parameters

Component	Parameter	Value [unit]
Clock Source J2	ped	8m [S]
	del	0 [S]
	periodical	1.0
Clock Source CLK2	ped	4m [S]
	del	0 [S]
	periodical	1.0
Clock Source K2	ped	16m [S]
	del	0 [S]
	periodical	1.0
Clock Source CLR2	ped	10m [S]
	del	15m [S]
	periodical	0
Clock Source PST2	ped	8m [S]
	del	20m [S]
	periodical	0
JK-Flip-Flop with Preset and Clear ffjkcp1	t_prop	0.0 [S]
	clk	CLK2.val
	j	J2.val

	k	K2.val
	clr	CLR2.val
	pst	PST2.val

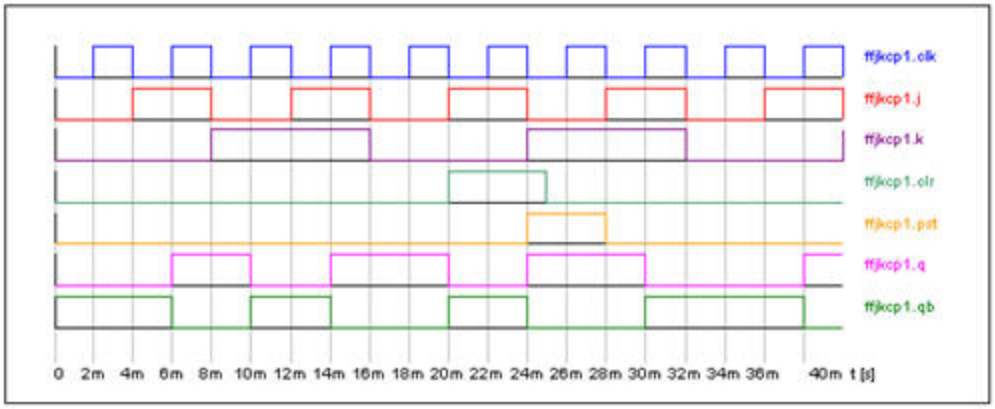

Figure 3. Simulation results-input and output of JK-Flip-Flop with Preset and Clear

Top

References

JK-Flip-Flop with Active-Low Preset and Clear

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

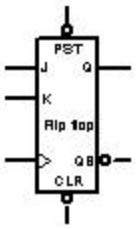

Figure 1. Component symbol

Description

The component represents a positive-edge triggered basic JK Flip-Flop with active-low preset and clear.

Top

Assumptions and Limitations

Top

Mathematical Description

Table 1. Truth Table

CLK	CLR	PST	J	K	Q	QB
0	1	1	J	K	NC	NC
X	0	1	J	K	0	1
X	1	0	J	K	1	0
rising	1	1	0	0	NC	NC
rising	1	1	0	1	0	1'
rising	1	1	1	0	1	0

rising	1	1	1	1	Q'	QB'
NC=No Change						
X=Don't Care						

Top

Netlist Syntax

COUPL ffjkcpal ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (t_prop := @t_prop , clk := @clk , j := @j , k := @k , clr := @clr , pst := @pst) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
clk	Clock input, data transferred at positive edge	bit	'0'
jin	J Input		'0'
kin	K Input	bit	'0'
pst	Asynchronous Active-Low Pre-set Signal	bit	'1'
clr	Asynchronous Active-Low Clear Signal	bit	'1'
t_prop	Propagation Time	real	0.0 [s]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
q	Data Output	Output	bit
qb	Complemented Data Output	Output	bit

Top

Example

This example illustrates the use of the JK-Flip-Flop with Active-Low Preset and Clear.

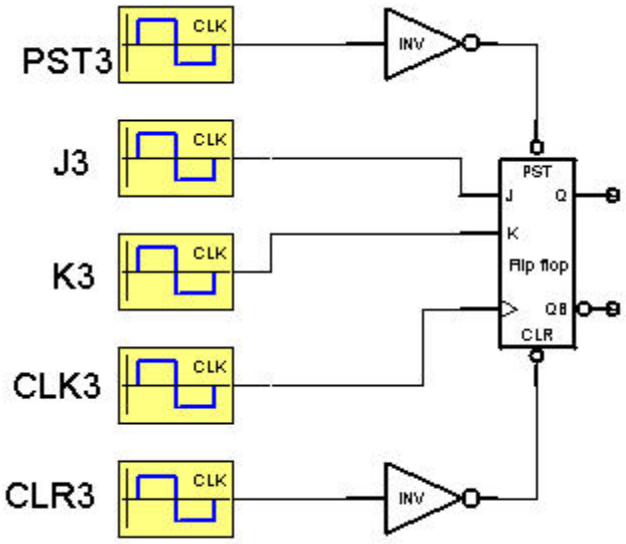

Figure 2. Application example of the JK-Flip-Flop with Active-Low Preset and Clear

Table 4. System Parameters

Component	Parameter	Value [unit]
Clock Source J3	ped	8m [S]
	del	0 [S]
	periodical	1.0
Clock Source CLK3	ped	4m [S]
	del	0 [S]
	periodical	1.0
Clock Source K3	ped	16m [S]
	del	0 [S]
	periodical	1.0
Clock Source CLR3	ped	10m [S]
	del	15m [S]
	periodical	0
Clock Source PST3	ped	8m [S]
	del	20m [S]
	periodical	0
Inverter inv1	tp_lh	0 [S]
	tp_hl	0 [S]
	x	CLR3.val
Inverter inv2		

	tp_lh	0 [S]
	tp_hl	0 [S]
	x	PST3.val
JK-Flip-Flop with Active-Low Preset and Clear ffjkcpal1	t_prop	0.0 [S]
	clk	CLK3.val
	j	J3.val
	k	K3.val
	clr	inv1.y
	pst	inv2.y

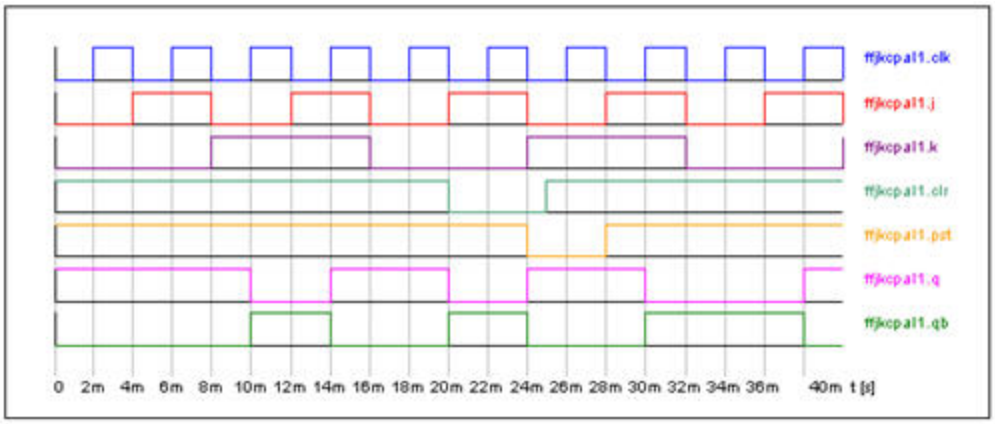

Figure 3. Simulation results-input and output of JK-Flip-Flop with Active-Low Preset and Clear

Top

References

T-Flip Flops

- Basic T-(Toggle) Flip-Flop (fft)
- T-Flip-Flop with asynchronous preset and clear (fftcp)
- T-Flip-Flop with asynchronous active-low preset and clear (fftcpal)

Basic T (Toggle) Flip-Flop

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

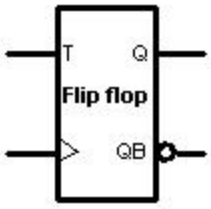

Figure 1. Component symbol

Description

The component represents a positive-edge triggered basic T (toggle) Flip-Flop.

Top

Assumptions and Limitations

Top

Mathematical Description

Table 1. Truth Table

CLK	T	Q	QB
0	T	NC	NC
rising	0	NC	NC
rising	1	Q'	QB'
NC=No Change			

Top

Netlist Syntax

COUPL fft ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (t_prop := @t_prop , clk := @clk , tin := @tin) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:-:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
clk	Clock input, data transferred at positive edge	bit	'0'
tin	Toggle Input	bit	'0'
t_prop	Propagation Time	real	0.0 [s]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
q	Data Output	Output	bit
qb	Complemented Data Output	Output	bit

Top

Example

This example illustrates the use of the T-Flip-Flop.

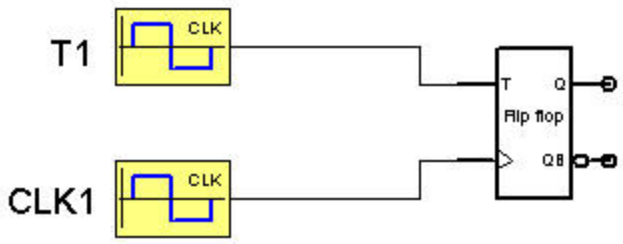

Figure 2. Application example of the T-Flip-Flop

Table 4. System Parameters

Component	Parameter	Value [unit]
Clock Source T1		

	ped	20m [S]
	del	0 [S]
	periodical	1.0
Clock Source CLK1	ped	4m [S]
	del	0 [S]
	periodical	1.0
T-Flip-Flop fft1	t_prop	0.0 [S]
	clk	CLK1.val
	tin	T1.val

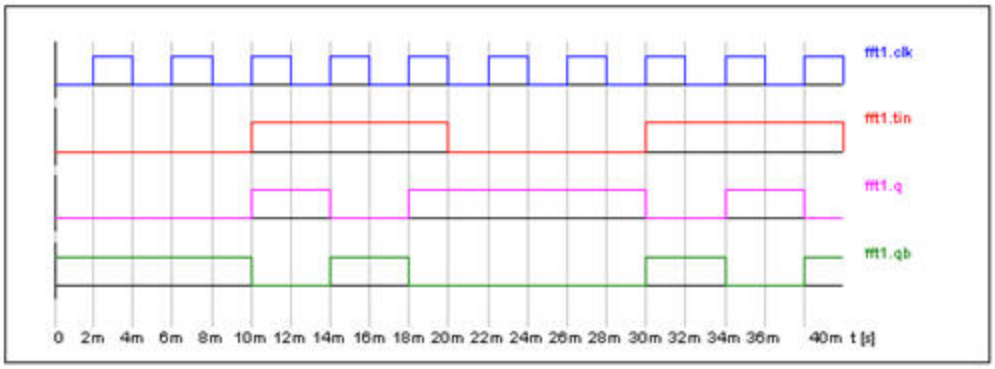

Figure 3. Simulation results-input and output of T-Flip-Flop

Top

References

T-Flip-Flop with Preset and Clear

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

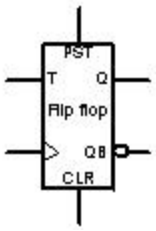

Figure 1. Component symbol

Description

The component represents a positive-edge triggered basic T Flip-Flop with preset and clear.

Top

Assumptions and Limitations

Top

Mathematical Description

Table 1. Truth Table

CLK	CLR	PST	T	Q	QB
0	0	0	T	NC	NC
X	1	0	T	0	1
X	0	1	T	1	0
1	0	0	0	NC	NC
1	0	0	1	Q'	QB'
NC=No Change					
X=Don’t Care					

Top

Netlist Syntax

COUPL fftcp ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (t_prop := @t_prop , clk := @clk , tin := @tin , pst := @pst , clr := @clr) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
clk	Clock input, data transferred at positive edge	bit	'0'
tin	Toggle Input	bit	'0'
pst	Asynchronous Active-High Pre-set Signal	bit	'0'
clr	Asynchronous Active-High Clear Signal	bit	'0'
t_prop	Propagation Time	real	0.0 [s]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
q	Data Output	Output	bit
qb	Complemented Data Output	Output	bit

Top

Example

This example illustrates the use of the T-Flip-Flop with Preset and Clear.

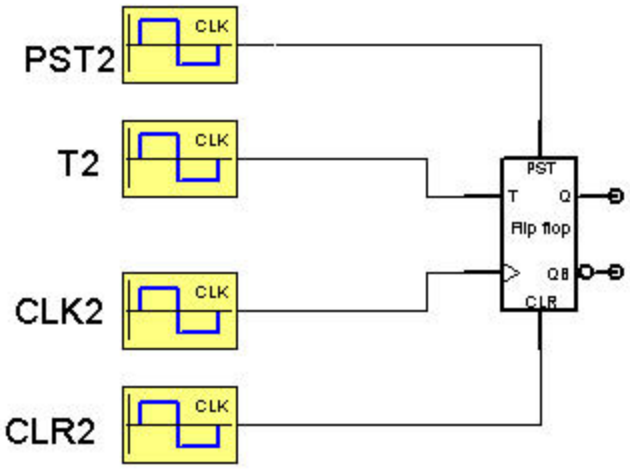

Figure 2. Application example of the T-Flip-Flop with Preset and Clear

Table 4. System Parameters

Component	Parameter	Value [unit]
Clock Source T2	ped	20m [S]
	del	0 [S]
	periodical	1.0
Clock Source CLK2	ped	4m [S]
	del	0 [S]
	periodical	1.0
Clock Source CLR2	ped	10m [S]
	del	15m [S]
	periodical	0
Clock Source PST2	ped	8m [S]
	del	20m [S]

	periodical	0
T-Flip-Flop with Preset and Clear fftcp1	t_prop	0.0 [S]
	clk	CLK2.val
	tin	T2.val
	clr	CLR2.val
	pst	PST2.val

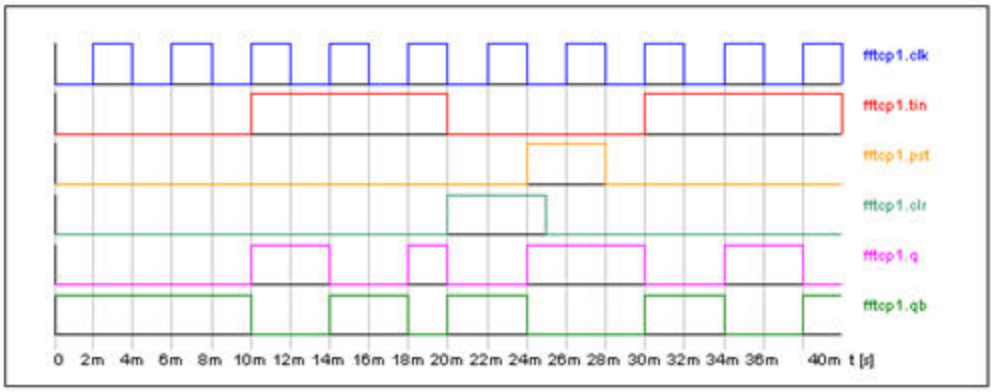

Figure 3. Simulation results-input and output of T-Flip-Flop with Preset and Clear

Top

References

T-Flip-Flop with Active-Low Preset and Clear

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

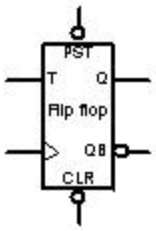

Figure 1. Component symbol

Description

The component represents a positive-edge triggered basic T Flip-Flop with active-low preset and clear.

Top

Assumptions and Limitations

Top

Mathematical Description

Table 1. Truth Table

CLK	CLR	PST	T	Q	QB
0	1	1	T	NC	NC
X	0	1	T	0	1
X	1	0	T	1	0
1	1	1	0	NC	NC
1	1	1	1	Q'	QB'
NC=No Change					
X=Don’t Care					

Top

Netlist Syntax

COUPL fftcpal ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (t_prop := @t_prop , clk := @clk , tin := @tin , pst := @pst , clr := @clr) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
clk	Clock input, data transferred at positive edge	bit	'0'
tin	Toggle Input	bit	'0'
pst	Asynchronous Active-Low Pre-set Signal	bit	'1'
clr	Asynchronous Active-Low Clear Signal	bit	'1'
t_prop	Propagation Time	real	0.0 [s]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
q	Data Output	Output	bit
qb	Complemented Data Output	Output	bit

Top

Example

This example illustrates the use of the T-Flip-Flop with Active-Low Preset and Clear.

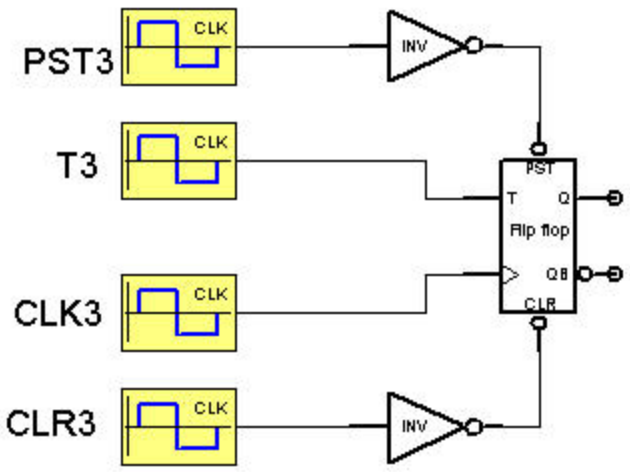

Figure 2. Application example of the T-Flip-Flop with Active-Low Preset and Clear

Table 4. System Parameters

Component	Parameter	Value [unit]
Clock Source T3	ped	20m [S]
	del	0 [S]
	periodical	1.0
Clock Source CLK3	ped	4m [S]
	del	0 [S]
	periodical	1.0
Clock Source CLR3	ped	10m [S]
	del	15m [S]
	periodical	0
Clock Source PST3	ped	8m [S]
	del	20m [S]

	periodical	0
Inverter inv1	tp_lh	0 [S]
	tp_hl	0 [S]
	x	CLR3.val
Inverter inv2	tp_lh	0 [S]
	tp_hl	0 [S]
	x	PST3.val
T-Flip-Flop with Active-Low Preset and Clear fftcp1	t_prop	0.0 [S]
	clk	CLK3.val
	tin	T3.val
	clr	inv1.y
	pst	inv2.y

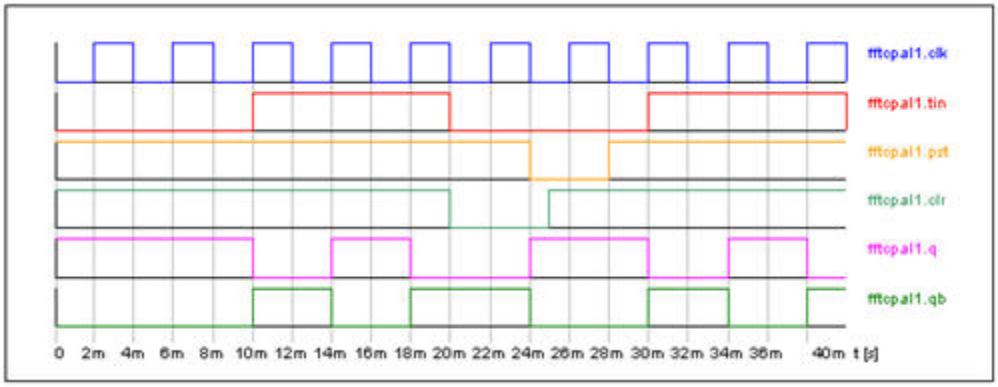

Figure 3. Simulation results-input and output of T-Flip-Flop with Active-Low Preset and Clear

Top

References

Latches

The latches can be used in simple sequential digital circuit simulations with propagation delays. The models operate on digital signals of type **BIT**. Delays are specified in terms of propagation delays (**t_prop**).

All models are positive edge triggered and some have the capability of having the outputs preset to '1' or cleared to '0' through asynchronous **PST** and **CLR** inputs. If both **PST** and **CLR** inputs are active at the same time, priority is given to the **PST** input.

- Basic D-Latch (ld)
- D-Latch with asynchronous preset and clear (ldcp)
- D Latch with asynchronous active-low preset and clear (ldcpal)
- Basic SR-Latch (lsr)

Basic D-Latch

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

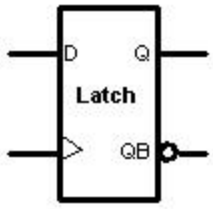

Figure 1. Component symbol

Description

The component represents a positive-edge triggered basic D-Latch.

Top

Assumptions and Limitations

Top

Mathematical Description

Table 1. Truth Table

CLK	D	Q	QB
0	D	NC	NC
1	0	0	1
1	1	1	0
NC=No Change			

Top

Netlist Syntax

COUPL Id ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (t_prop := @t_prop , clk := @clk , din := @din) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:-:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
clk	Clock input, data transferred at positive edge	bit	'0'
din	Data Input	bit	'0'
t_prop	Propagation Time	real	0.0 [s]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
q	Data Output	Output	bit
qb	Complemented Data Output	Output	bit

Top

Example

This example illustrates the use of the Basic D-Latch.

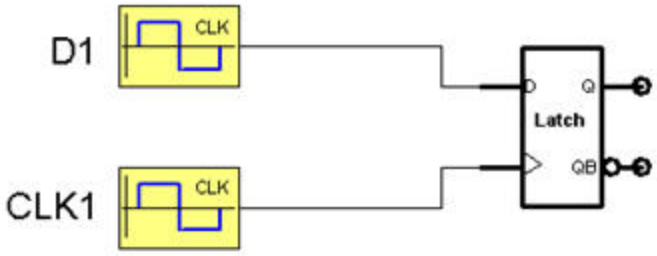

Figure 2. Application example of the Basic D-Latch

Table 4. System Parameters

Component	Parameter	Value [unit]
Clock Source D1	ped	12m [S]

	del	0 [S]
	periodical	1.0
Clock Source CLK1	ped	8m [S]
	del	0 [S]
	periodical	1.0
Basic D-Latch ld1	t_prop	0.0 [S]
	clk	CLK1.val
	din	D1.val

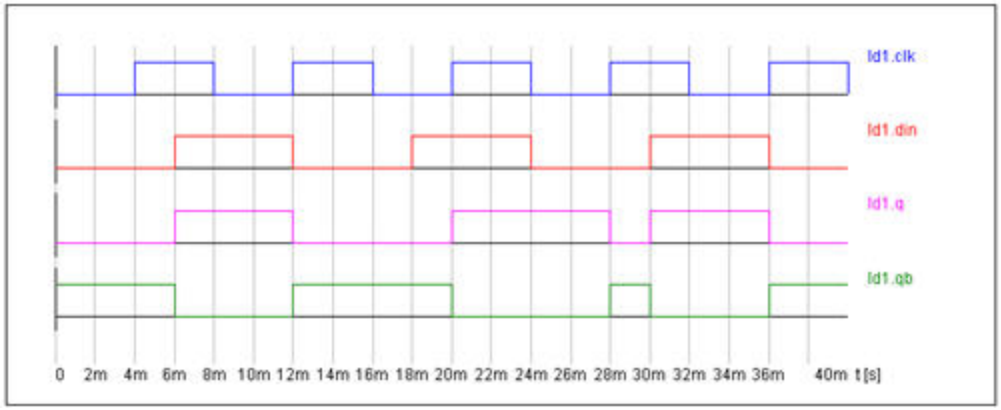

Figure 3. Simulation results-input and output of Basic D-Latch

Top

References

Basic D-Latch with Preset and Clear

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

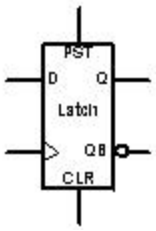

Figure 1. Component symbol

Description

The component represents a positive-edge triggered basic D-Latch with preset and clear.

Top

Assumptions and Limitations

Top

Mathematical Description

Table 1. Truth Table

CLK	CLR	PST	D	Q	QB
0	0	0	D	NC	NC
X	1	0	D	0	1
X	0	1	D	1	0
1	0	0	0	0	1
1	0	0	1	1	0
NC=No Change					
X=Don’t Care					

Top

Netlist Syntax

COUPL ldcp ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (t_prop := @t_prop , clk := @clk , din := @din , clr := @clr , pst := @pst) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
clk	Clock input, data transferred at positive edge	bit	'0'
din	Data Input	bit	'0'
pst	Asynchronous Active-High Pre-set Signal	bit	'0'
clr	Asynchronous Active-High Clear Signal	bit	'0'
t_prop	Propagation Time	real	0.0 [s]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
q	Data Output	Output	bit
qb	Complemented Data Output	Output	bit

Top

Example

This example illustrates the use of the D-Latch with Preset and Clear.

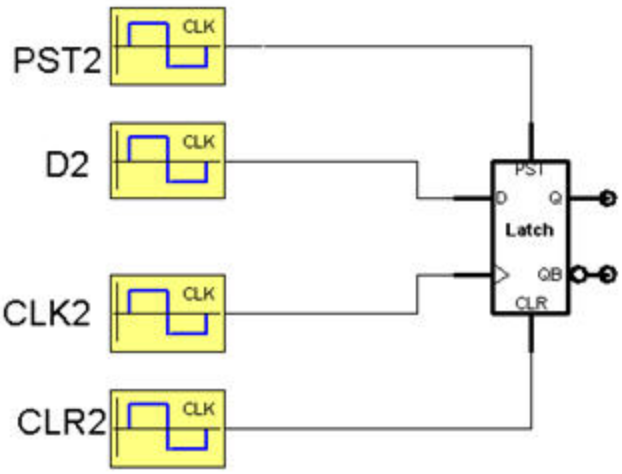

Figure 2. Application example of the D-Latch with Preset and Clear

Table 4. System Parameters

Component	Parameter	Value [unit]
Clock Source D2	ped	10m [S]
	del	0 [S]
	periodical	1.0
Clock Source CLK2	ped	4m [S]
	del	0 [S]
	periodical	1.0
Clock Source CLR2	ped	10m [S]
	del	15m [S]
	periodical	0
Clock Source PST2	ped	8m [S]
	del	20m [S]

	periodical	0
D-Latch with Preset and Clear ldcp1	t_prop	0.0 [S]
	clk	CLK2.val
	din	D2.val
	clr	CLR2.val
	pst	PST2.val

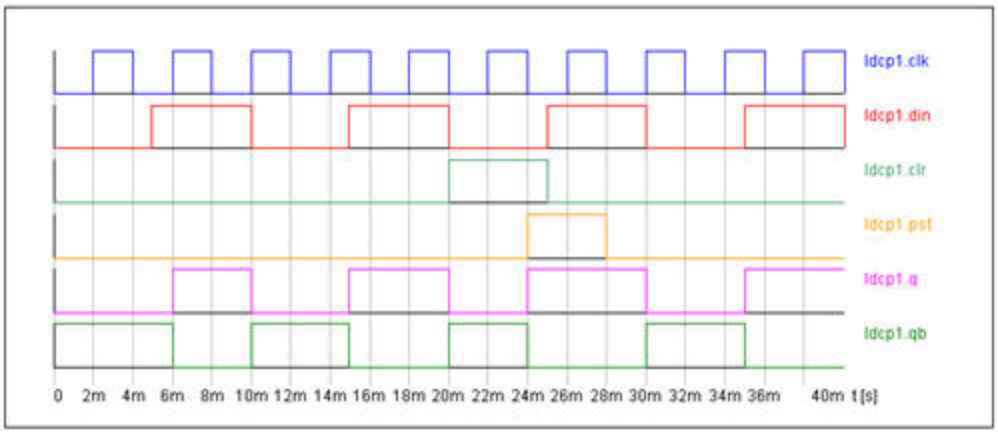

Figure 3. Simulation results-input and output of Basic D-Latch with Preset and Clear

Top

References

Basic D-Latch with Active-Low Preset and Clear

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

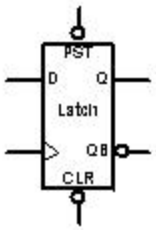

Figure 1. Component symbol

Description

The component represents a positive-edge triggered D-Latch with active-low preset and clear.

Top

Assumptions and Limitations

Top

Mathematical Description

Table 1. Truth Table

CLK	CLR	PST	D	Q	QB
0	1	1	D	NC	NC
X	0	1	D	0	1
X	1	0	D	1	0
1	1	1	0	0	1
1	1	1	1	1	0
NC=No Change					
X=Don’t Care					

Top

Netlist Syntax

COUPL ldcpal ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (t_prop := @t_prop , clk := @clk , din := @din , clr := @clr , pst := @pst) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
clk	Clock input, data transferred at positive edge	bit	'0'
din	Data Input	bit	'0'
pst	Asynchronous Active-Low Pre-set Signal	bit	'1'
clr	Asynchronous Active-Low Clear Signal	bit	'1'
t_prop	Propagation Time	real	0.0 [s]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
q	Data Output	Output	bit
qb	Complemented Data Output	Output	bit

Top

Example

This example illustrates the use of the D-Latch with Active-Low Preset and Clear.

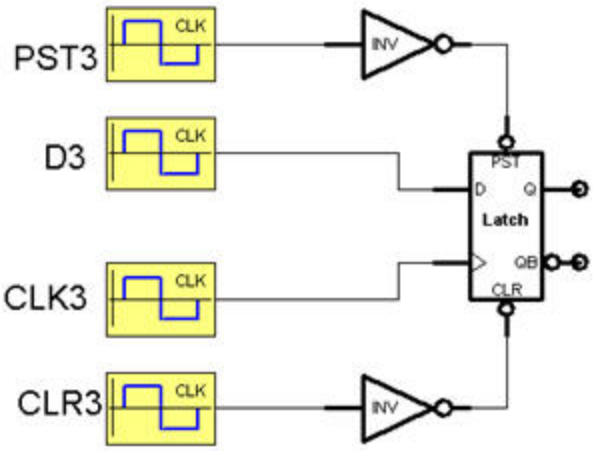

Figure 2. Application example of the D-Latch with Active-Low Preset and Clear

Table 4. System Parameters

Component	Parameter	Value [unit]
Clock Source D3	ped	10m [S]
	del	0 [S]
	periodical	1.0
Clock Source CLK3	ped	4m [S]
	del	0 [S]
	periodical	1.0
Clock Source CLR3	ped	10m [S]
	del	15m [S]
	periodical	0
Clock Source PST3	ped	8m [S]
	del	20m [S]
	periodical	0
Inverter inv1	tp_lh	0 [S]
	tp_hl	0 [S]
	x	CLR3.val
Inverter inv2	tp_lh	0 [S]
	tp_hl	0 [S]
	x	PST3.val

D-Latch with Active-Low Preset and Clear ldcpal1	t_prop	0.0 [S]
	clk	CLK3.val
	din	D3.val
	clr	inv1.y
	pst	inv2.y

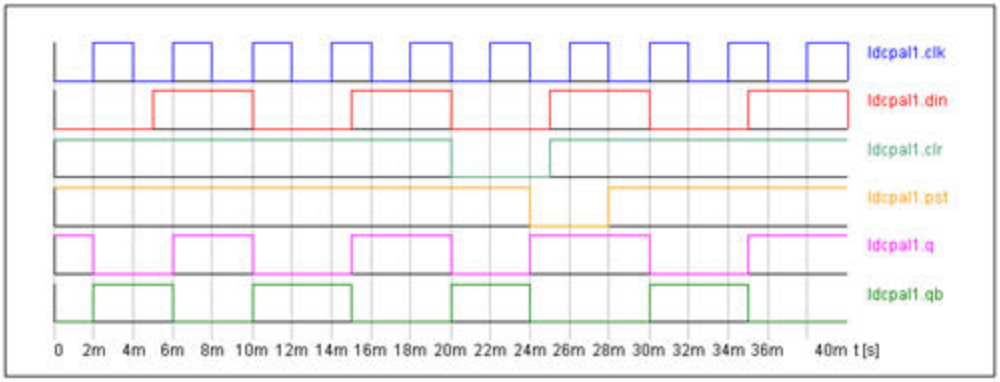

Figure 3. Simulation results-input and output of D-Latch with Active-Low Preset and Clear

Top

References

SR Latch Based on NOR Logic

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

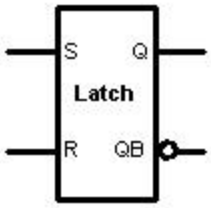

Figure 1. Component symbol

Description

The macro represents a SR Latch based on NOR logic.

Top

Assumptions and Limitations

Top

Mathematical Description

Table 1. Truth Table

S	R	Q	QB
0	0	NC	NC
0	1	0	1
1	0	1	0
1	1	UD	UD
NO=No Change, UD=Undefined			

Top

Netlist Syntax

COUPL lsr ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (t_prop := @t_prop , s := @s , r := @r) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
s	Set Input	bit	'0'
r	Reset Input	bit	'0'
t_prop	Propagation Time	real	0.0 [s]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
q	Data Output	Output	bit
qb	Complemented Data Output	Output	bit

Top

Example

This example illustrates the use of the SR Latch Based on NOR Logic.

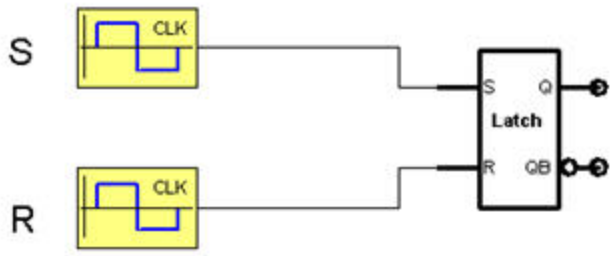

Figure 2. Application example of the SR Latch Based on NOR Logic

Table 4. System Parameters

Component	Parameter	Value [unit]
Clock Source S	ped	4m [S]

	del	0 [S]
	periodical	1.0
Clock Source R	ped	8m [S]
	del	0 [S]
	periodical	1.0
SR Latch lsr1	t_prop	0.5m [S]
	s	S.val
	r	R.val

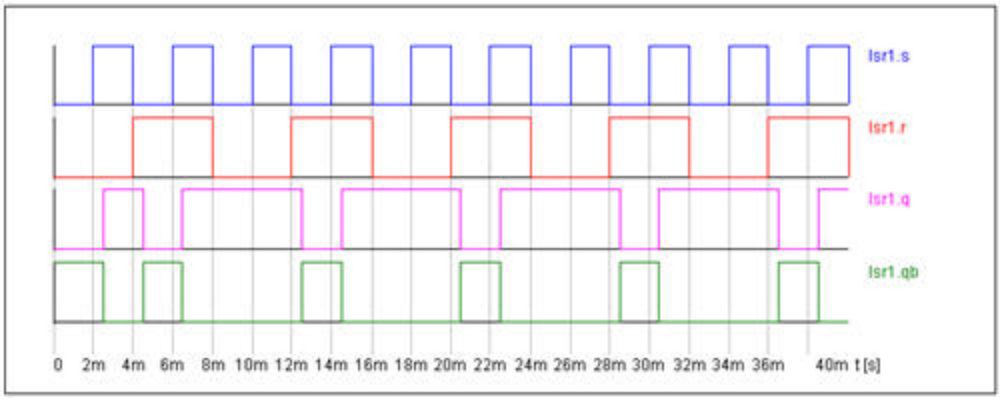

Figure 3. Simulation results-input and output of SR Latch

Top

References

Logic Blocks

The logic blocks can be used to simulate simple digital circuits for combinational circuits with delays. The models operate on digital signals of type **BIT**. Delays are expressed in terms of rise time delay (**tp_lh**) and fall time delay (**tp_hl**).

Two-Bit Comparator

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

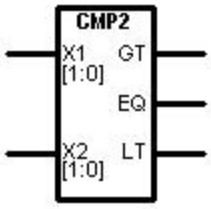

Figure 1. Component symbol

Description

The component represents a two-bit comparator. The comparator operates on digital signals of type **BIT**. It compares the two input two-bit vector values and indicates the result with a '1' on one of the three outputs, *greater than, less than, or equal to* lines. Delays are specified in terms of propagation delays (**t_prop**).

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

COUPL cmp2 ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (t_prop := @t_prop , x1 := @x1 , x1[1] := @x1[1], x1[0] := @x1[0], x2 := @x2 , x2[1] := @x2[1], x2[0] := @x2[0]) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:-:=\\"@Architecture\\"); ;

Top

Parameters

Table 1

Name	Description	Data Type	Default Value [Unit]
x1	Two-bit Input Signal Vector X1	bit_vector	'0'
x2	Two-bit Input Signal Vector X2	bit_vector	'0'
t_prop	Propagation Time	real	0.0 [s]

Top

Input/Output Quantities

Table 2

Name	Description [Unit]	Direction	Data Type
lto	Less than Output (X1<X2)	Output	bit
eqo	Equal to Output (X1=X2)	Output	bit
gto	Greater than Output (X1>X2)	Output	bit

Top

Example

See Two-Bit Comparator Example

References

Two-Bit Comparator Example

This example illustrates the use of the Two-Bit Comparator. Inputs to the comparator are supplied by the Two-Bit Vector Source.

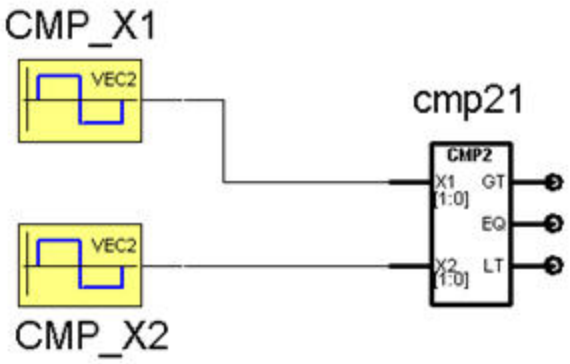

Figure 1. Application example of the Two-Bit Comparator

Table 1. System Parameters

Component	Parameter	Value [unit]
Two-Bit Vector Source CMP_X1	ped	4m [S]
	del	10m [S]
Two-Bit Vector Source CMP_X2	ped	5m [S]
	del	0 [S]
Comparator cmp21	t_prop	0 [S]
	x1	CMP_X1.val
	x2	CMP_X2.val

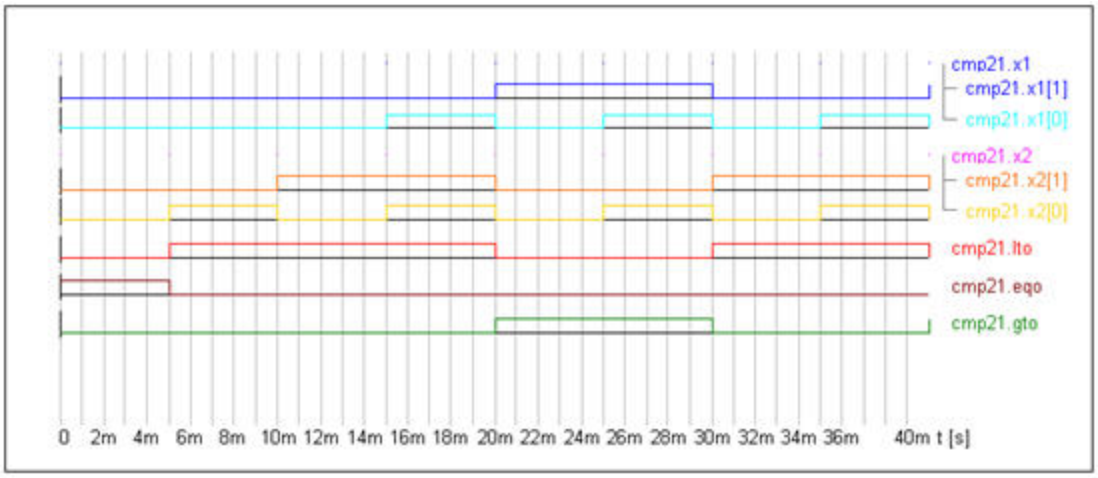

Figure 2. Simulation results-input and output of the Two-Bit Comparator

Top

References

2-to-4 Line Decoder

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

Figure 1. Component symbol

Description

The component represents a two-to-four line decoder. The decoder operates on digital signals of type **BIT**. Depending on the coded value available in the two-bit input line, the decoder model indicates a result of '1' in one of the four output lines. Delays are specified in terms of propagation delays (**t_prop**).

Top

Assumptions and Limitations

Top

Mathematical Description

Table 1. Truth Table

X0	X1	Y0	Y1	Y2	Y3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Top

Netlist Syntax

COUPL dec2_4 ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (t_prop := @t_prop , x0 := @x0 , x1 := @x1) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:-:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
x0	Input Signal X1	bit	'0'
x1	Input Signal X2	bit	'0'
t_prop	Propagation Time	real	0.0 [s]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
y0	Output Y1	Output	bit
y1	Output Y2	Output	bit
y2	Output Y3	Output	bit
y3	Output Y4	Output	bit

Top

Example

See 2-to-4 Line Decoder Example

Top

References

2-to-4 Line Decoder Example

This example illustrates the use of the 2-to-4 Line Decoder. Inputs to the decoder are supplied with Clock Sources.

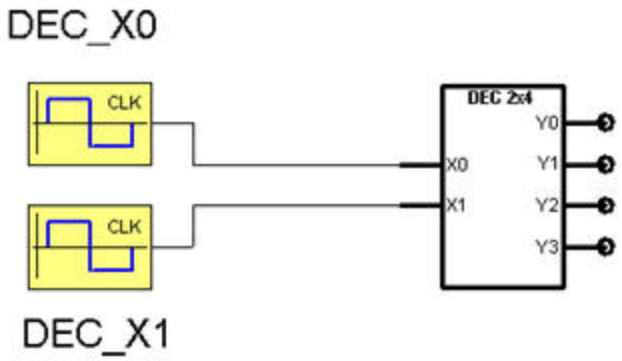

Figure 1. Application example of the 2-to-4 Line Decoder

Table 1. System Parameters

Component	Parameter	Value [unit]
Clock Source DEC_X0	ped	15m [S]
	del	0 [S]
	periodical	1.0 [S]
Clock Source DEC_X1	ped	30m [S]
	del	0 [S]
	periodical	1.0 [S]
Decoder(2x4) dec2_41	t_prop	0 [S]
	x0	DEC_X0.val
	x1	DEC_X1.val

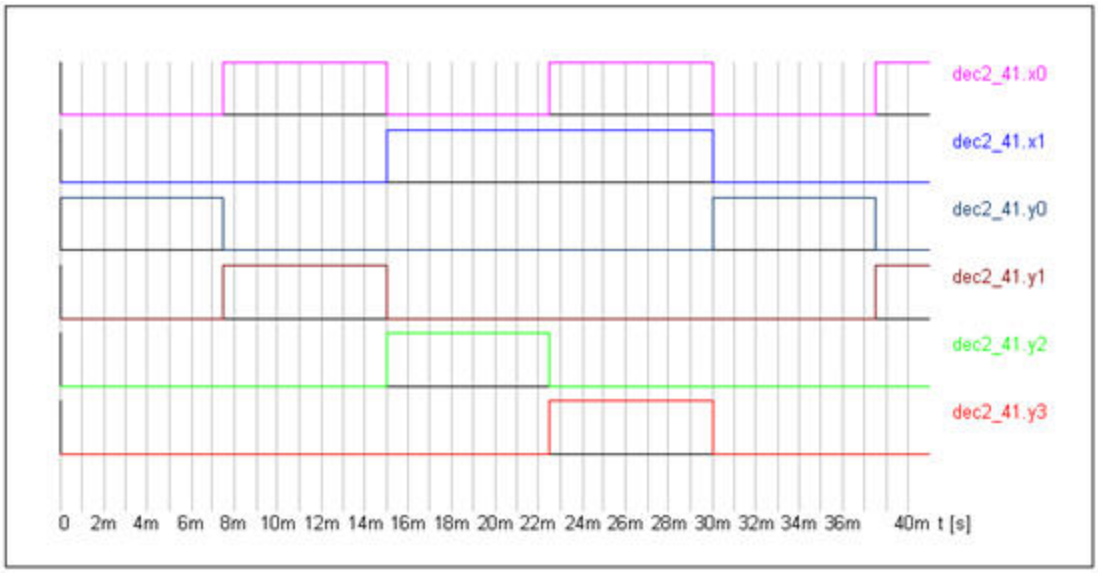

Figure 2. Simulation results-input and output of the 2-to-4 Line Decoder

2-to-1 Multiplexer

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

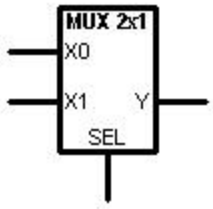

Figure 1. Component symbol

Description

This component represents a two-by-one multiplexer. The multiplexer operates on digital signals of type **BIT**. Depending on the input through the select line, one of the two inputs is transferred to the output. Delays are specified in terms of propagation delays (**t_prop**).

Top

Assumptions and Limitations

Top

Mathematical Description

Table 1. Truth Table

SEL	Y
0	X0
1	X1

Top

Netlist Syntax

COUPL mux2_1 ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (t_prop := @t_prop , x0 := @x0 , x1 := @x1 , sel := @sel) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
x0	Input Signal X0	bit	'0'
x1	Input Signal X1	bit	'0'
sel	Select Signal	bit	'0'
t_prop	Propagation Time	real	0.0 [s]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
y	Output Y	Output	bit

Top

Example

This example illustrates the use of the 2-to-1 Multiplexer.

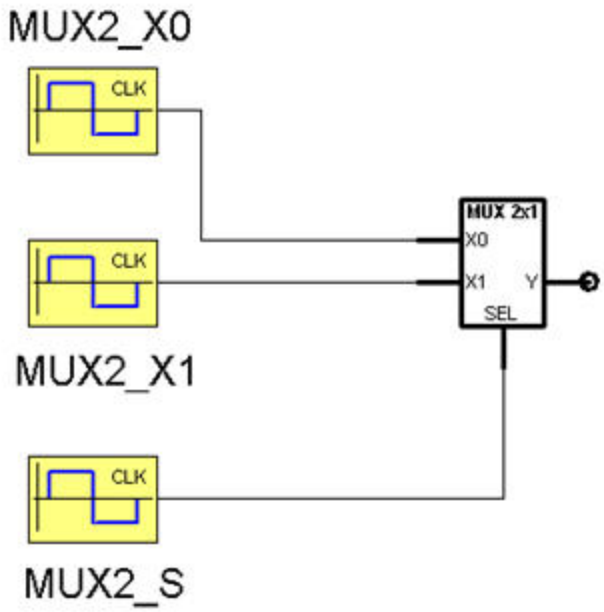

Figure 2. Application example of the 2-to-1 Multiplexer

Table 4. System Parameters

Component	Parameter	Value [unit]
Clock Source MUX2_X0	ped	2m [S]
	del	0 [S]
	periodical	1.0 [S]
Clock Source MUX2_X1	ped	10m [S]
	del	0 [S]
	periodical	1.0 [S]
Clock Source MUX2_S	ped	20m [S]
	del	0 [S]
	periodical	1.0 [S]
Multiplexer(2x1) mux2_11	t_prop	0 [S]
	x0	MUX2_X0.val
	x1	MUX2_X1.val
	sel	MUX2_S.val

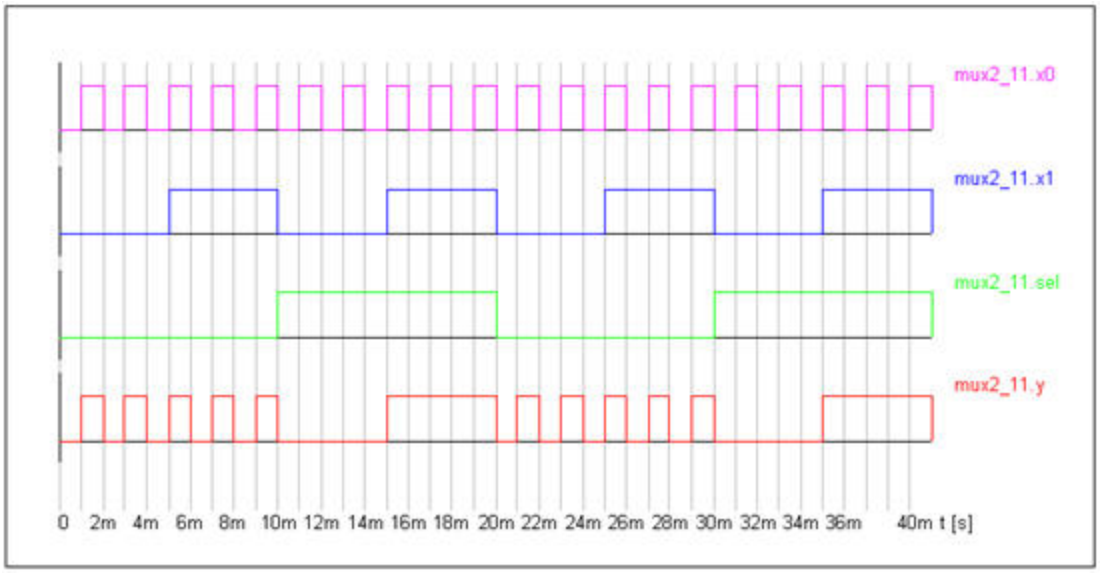

Figure 3. Simulation results-input and output of the 2-to-1 Multiplexer

Top

References

4-to-1 Multiplexer

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

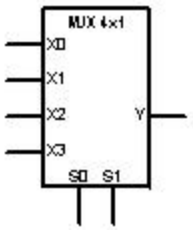

Figure 1. Component symbol

Description

This component represents a four-by-one multiplexer. The multiplexer operates on digital signals of type **BIT**. Depending on the input through the two select lines, one of the four inputs is transferred to the output. Delays are specified in terms of propagation delays (**t_prop**).

Top

Assumptions and Limitations

Top

Mathematical Description

Table 1. Truth Table

S0	S1	Y
0	0	X1
0	1	X2
1	0	X3
1	1	X4

Top

Netlist Syntax

COUPL mux4_1 ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (t_prop := @t_prop , x0 := @x0 , x1 := @x1 , x2 := @x2 , x3 := @x3 , sel := @sel , sel[1] := @sel[1], sel[0] := @sel[0]) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:-:=\\"@Architecture\\"); ;

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
x0	Input Signal X1	bit	'0'
x1	Input Signal X2	bit	'0'
x2	Input Signal X3	bit	'0'
x3	Input Signal X4	bit	'0'
sel	Select Vector	bit	'0'
t_prop	Propagation Time	real	0.0 [s]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
y	Output Y	Output	bit

Top

Example

This example illustrates the use of the 4-to-1 Multiplexer.

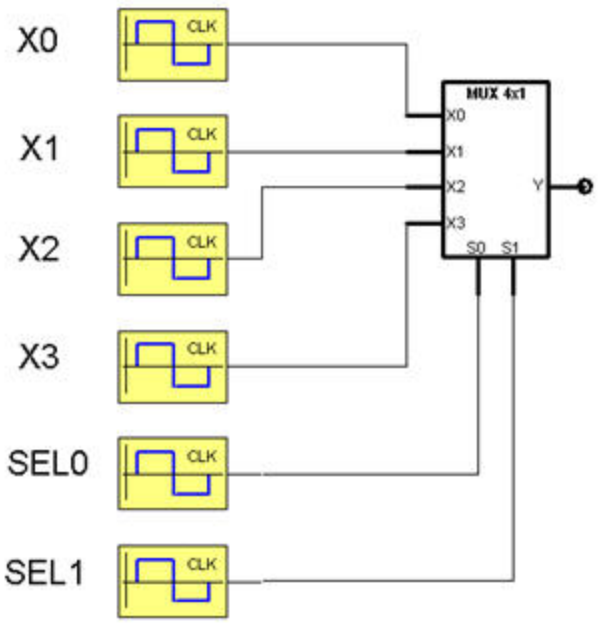

Figure 2. Application example of the 4-to-1 Multiplexer

Table 4. System Parameters

Component	Parameter	Value [unit]
Clock Source X0	ped	0.5m [S]
	del	0 [S]
	periodical	1.0 [S]
Clock Source X1	ped	1m [S]
	del	0 [S]
	periodical	1.0 [S]
Clock Source X2	ped	2m [S]
	del	0 [S]
	periodical	1.0 [S]
Clock Source X3		

	ped	4m [S]
	del	0 [S]
	periodical	1.0 [S]
Clock Source SEL0	ped	20m [S]
	del	0 [S]
	periodical	1.0 [S]
Clock Source SEL1	ped	40m [S]
	del	0 [S]
	periodical	1.0 [S]
Multiplexer(4x1) mux4_11	t_prop	0 [S]
	x0	X0.val
	x1	X1.val
	x2	X2.val
	x3	X3.val
	sel(1)	SEL1.val
	sel(0)	SEL0.val

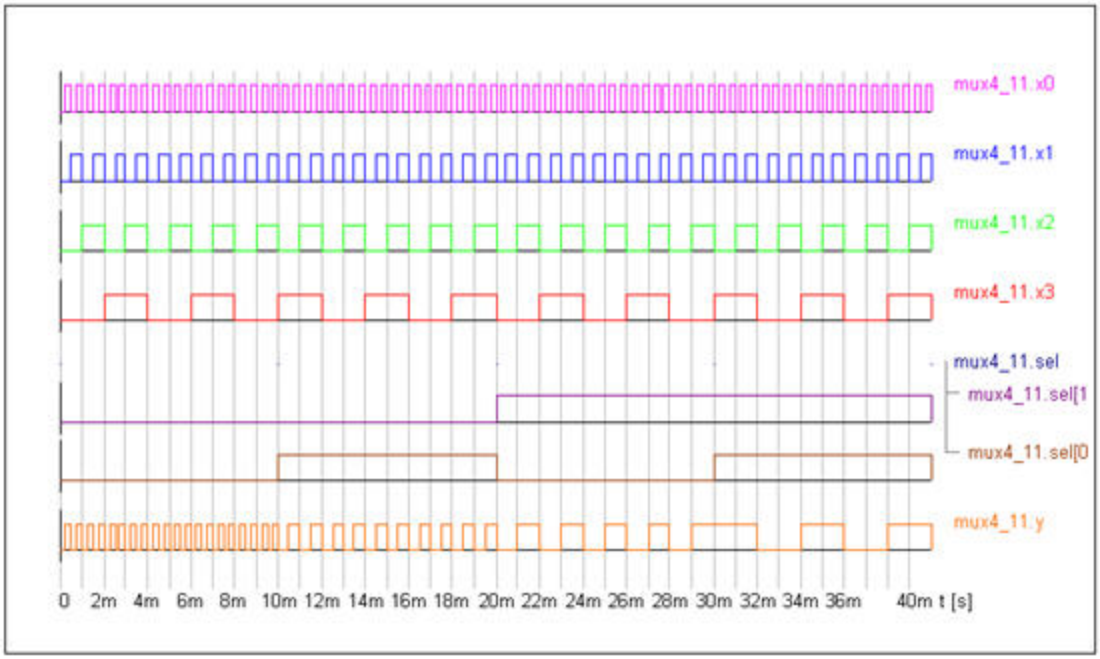

Figure 3. Simulation results-input and output of the 4-to-1 Multiplexer

Top

References

Four Bit Bidirectional Serial Shift Register

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

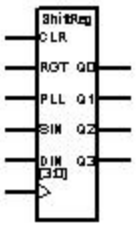

Figure 1. Component symbol

Description

This component represents a 4-bit bidirectional serial shift register with asynchronous clear (positive edge triggered) with parallel load. The register operates on digital signals of type **BIT**. Delays are specified in terms of propagation delays (**t_prop**). The model can shift four bits of data serially, either to the right (right input = '1') or to the left (right input = '0'). All four bits of data can be loaded in the model in parallel (pll = '1', use d_in input)) or in serial (pll = '0', use s_in input). It is also possible to clear the data in the model using the asynchronous, active-high clear input.

Top

Assumptions and Limitations

Top

Mathematical Description

Table 1. Truth Table

CLK	CLR	DAT	SL0SR1	Q3	Q2	Q1	Q0
0	0	1	1	NC	NC	NC	NC
0	1	1	1	0	0	0	0

1	0	D1	0	D1	Q3	Q2	Q1
1	0	D2	0	D2	D1	Q3	Q2
1	0	D3	0	D3	D2	D1	Q3
1	0	D4	0	D4	D3	D2	D1
1	0	D5	1	D3	D2	D1	D5
1	0	D6	1	D2	D1	D5	D6
1	0	D7	1	D1	D5	D6	D7
1	0	D8	1	D5	D6	D7	D8
NC=No Change							

Top

Netlist Syntax

COUPL sr4 ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (t_prop := @t_prop , clk := @clk , clr := @clr , pll := @pll , d_in := @d_in , d_in[3] := @d_in[3], d_in[2] := @d_in[2], d_in[1] := @d_in[1], d_in[0] := @d_in[0], s_in := @s_in , right := @right) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
clk	Clock	bit	'0'
clr	Asynchronous Active-High Clear	bit	'0'
pll	Parallel/Serial Data Load	bit	'0'
right	Right Direction of Serial Shift	bit	'0'
s_in	Serial Data Input	bit	'0'
d_in	Parallel Data Input Vector	bit	'0'
t_prop	Propagation Time	real	0.0 [s]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
q	4-Bit Output Vector	Output	bit

Top

Example

See Four Bit Bidirectional Serial Shift Register Example

Top

References

Four Bit Bidirectional Serial Shift Register Example

This example illustrates the use of the Four Bit Bidirectional Serial Shift Register. Inputs to the shift register are supplied by Clock Sources and a Four-Bit Vector Source.

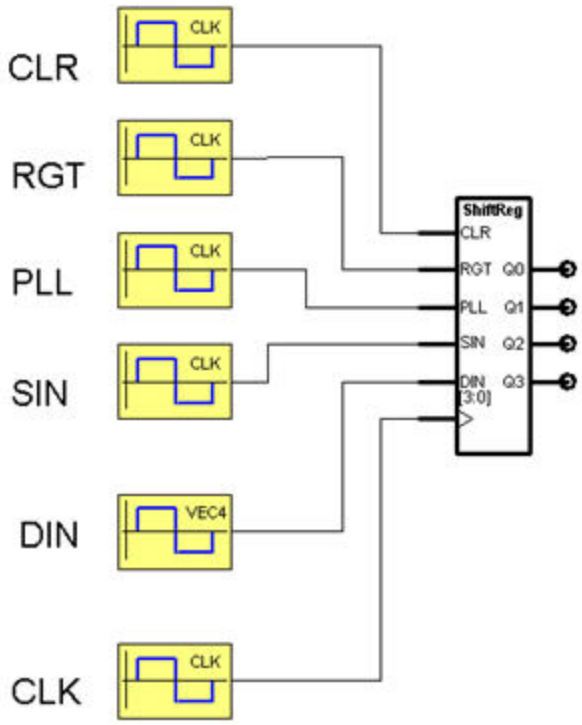

Figure 1. Application example of the Four Bit Bidirectional Serial Shift Register

Table 1. System Parameters

Component	Parameter	Value [unit]
Clock Source CLR	ped	5m [S]
	del	10m [S]
	periodical	0 [S]
Clock Source RGT	ped	20m [S]
	del	0 [S]
	periodical	0 [S]
Clock Source PLL	ped	25m [S]
	del	0 [S]
	periodical	1.0 [S]
Clock Source SIN	ped	6m [S]
	del	0 [S]
	periodical	1.0 [S]
Clock Source CLK	ped	2m [S]
	del	0 [S]
	periodical	1.0 [S]
Four-Bit Vector Source DIN	ped	1m [S]
	del	0 [S]
Shift Register sr41	t_prop	0 [S]

	clk	CLK.val
	clr	CLR.val
	pll	PLL.val
	d_in	DIN.val
	s_in	SIN.val
	right	RGT.val

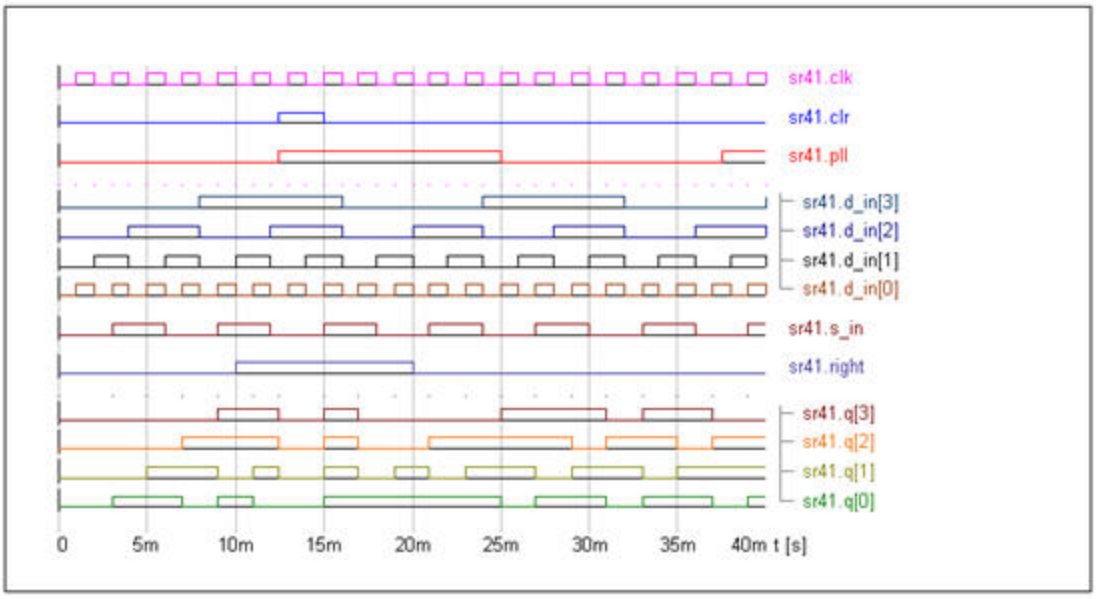

Figure 2. Simulation results-input and output of the Four Bit Bidirectional Serial Shift Register

Logic Gates

The logic gates can be used to simulate simple digital circuits for combinational circuits with propagation delays. The components operate on digital signals of type **BIT**. Delays are expressed in terms of rise time delay (**tp_lh**) and fall time delay (**tp_hl**).

Two Input AND Gate

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

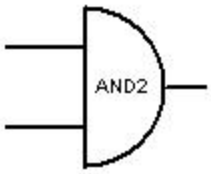

Figure 1. Component symbol

Description

The component represents a two input AND gate.

Top

Assumptions and Limitations

Top

Mathematical Description

The output Y can be expressed as:

Y=X1 AND X2

Table 1. Truth Table

X2	X1	Y
0	0	0
0	1	0
1	0	0
1	1	1

Top

Netlist Syntax

COUPL and2 ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (tp_lh := @tp_lh , tp_hl := @tp_hl , x1 := @x1 , x2 := @x2) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
x1	Input Signal X1	bit	'0'
x2	Input Signal X2	bit	'0'
tp_lh	Rise Time Delay	real	0.0 [s]
tp_hl	Fall Time Delay	real	0.0 [s]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
y	Output Y	Output	bit

Top

Example

See Logic Gates Example: AND & NAND

Top

References

Two Input AND Gate with One Inverted Input

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

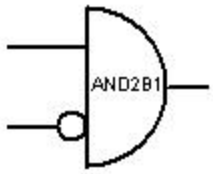

Figure 1. Component symbol

Description

The component represents a two input AND gate with inverted input X2.

Top

Assumptions and Limitations

Top

Mathematical Description

The output Y can be expressed as:

Y =X1 AND X2'

Table 1. Truth Table

X2	X1	Y
0	0	0
0	1	1
1	0	0
1	1	0

Top

Netlist Syntax

COUPL and2b1 ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (tp_lh := @tp_lh , tp_hl := @tp_hl , x1 := @x1 , x2 := @x2) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
x1	Input Signal X1	bit	'0'
x2	Input Signal X2	bit	'0'
tp_lh	Rise Time Delay	real	0.0 [s]
tp_hl	Fall Time Delay	real	0.0 [s]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
y	Output Y	Output	bit

Top

Example

See Logic Gates Example: AND & NAND

Top

References

Three Input AND Gate

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

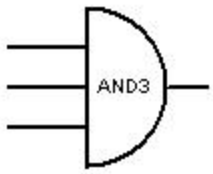

Figure 1. Component symbol

Description

The component represents a three input AND gate.

Top

Assumptions and Limitations

Top

Mathematical Description

The output Y can be expressed as:

Y =X1 AND X2 AND X3

Table 1. Truth Table

X3	X2	X1	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0

1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Top

Netlist Syntax

COUPL and3 ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (tp_lh := @tp_lh , tp_hl := @tp_hl , x1 := @x1 , x2 := @x2 , x3 := @x3) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
x1	Input Signal X1	bit	'0'
x2	Input Signal X2	bit	'0'
x3	Input Signal X3	bit	'0'
tp_lh	Rise Time Delay	real	0.0 [s]
tp_hl	Fall Time Delay	real	0.0 [s]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
y	Output Y	Output	bit

Top

Example

See Logic Gates Example: AND & NAND

Top

References

Four Input AND Gate

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

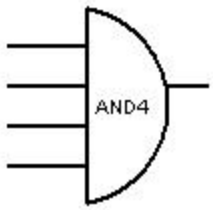

Figure 1. Component symbol

Description

The component represents a four input AND gate.

Top

Assumptions and Limitations

Top

Mathematical Description

The output Y can be expressed as:

Y =X1 AND X2 AND X3 AND X4

Table 1. Truth Table

X4	X3	X2	X1	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0

0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

Top

Netlist Syntax

COUPL and4 ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (tp_lh := @tp_lh , tp_hl := @tp_hl , x1 := @x1 , x2 := @x2 , x3 := @x3 , x4 := @x4) DST: SIM(Type:=SimVHDL) SRC: DB (File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
x1	Input Signal X1	bit	'0'
x2	Input Signal X2	bit	'0'
x3	Input Signal X3	bit	'0'
x4	Input Signal X4	bit	'0'
tp_lh	Rise Time Delay	real	0.0 [s]
tp_hl	Fall Time Delay	real	0.0 [s]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
y	Output Y	Output	bit

Top

Example

See Logic Gates Example: AND & NAND

Top

References

Inverter Gate

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

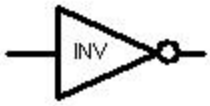

Figure 1. Component symbol

Description

The component represents a Inverter gate.

Top

Assumptions and Limitations

Top

Mathematical Description

Table 1. Truth Table

X1	Y
0	1
1	0

Top

Netlist Syntax

COUPL inv ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (tp_lh := @tp_lh , tp_hl := @tp_hl , x := @x) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:-:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
x	Input Signal X	bit	'0'
tp_lh	Rise Time Delay	real	0.0 [s]
tp_hl	Fall Time Delay	real	0.0 [s]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
y	Output Y	Output	bit

Top

Example

See Logic Gates Example: XOR & XNOR & INV

Top

References

Two Input NAND Gate

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

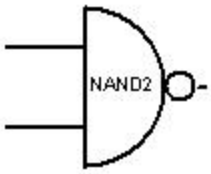

Figure 1. Component symbol

Description

The component represents a two input NAND gate.

Top

Assumptions and Limitations

Top

Mathematical Description

The output Y can be expressed as:

Y=NOT(X1 AND X2)

Table 1. Truth Table

X2	X1	Y
0	0	1
0	1	1
1	0	1
1	1	0

Top

Netlist Syntax

COUPL nand2 ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (tp_lh := @tp_lh , tp_hl := @tp_hl , x1 := @x1 , x2 := @x2) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
x1	Input Signal X1	bit	'0'
x2	Input Signal X2	bit	'0'
tp_lh	Rise Time Delay	real	0.0 [s]
tp_hl	Fall Time Delay	real	0.0 [s]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
y	Output Y	Output	bit

Top

Example

See Logic Gates Example: AND & NAND

Top

References

Two Input NAND Gate with One Inverted Input

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

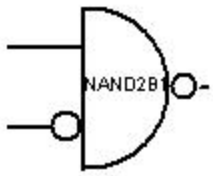

Figure 1. Component symbol

Description

The component represents a two input NAND gate with inverted input X2.

Top

Assumptions and Limitations

Top

Mathematical Description

The output Y can be expressed as:

Y=NOT(X1 AND X2')

Table 1. Truth Table

X2	X1	Y
0	0	1
0	1	0
1	0	1
1	1	1

Top

Netlist Syntax

COUPL nand2b1 ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (tp_lh := @tp_lh , tp_hl := @tp_hl , x1 := @x1 , x2 := @x2) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
x1	Input Signal X1	bit	'0'
x2	Input Signal X2	bit	'0'
tp_lh	Rise Time Delay	real	0.0 [s]
tp_hl	Fall Time Delay	real	0.0 [s]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
y	Output Y	Output	bit

Top

Example

See Logic Gates Example: AND & NAND

Top

References

Three Input NAND Gate

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

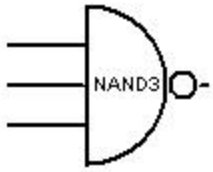

Figure 1. Component symbol

Description

The component represents a three input NAND gate.

Top

Assumptions and Limitations

Top

Mathematical Description

The output Y can be expressed as:

Y=NOT(X1 AND X2 AND X3)

Table 1. Truth Table

X3	X2	X1	Y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1

1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Top

Netlist Syntax

COUPL nand3 ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (tp_lh := @tp_lh , tp_hl := @tp_hl , x1 := @x1 , x2 := @x2 , x3 := @x3) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
x1	Input Signal X1	bit	'0'
x2	Input Signal X2	bit	'0'
x3	Input Signal X3	bit	'0'
tp_lh	Rise Time Delay	real	0.0 [s]
tp_hl	Fall Time Delay	real	0.0 [s]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
y	Output Y	Output	bit

Top

Example

See Logic Gates Example: AND & NAND

Top

References

Four Input NAND Gate

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

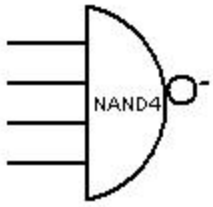

Figure 1. Component symbol

Description

The component represents a four input NAND gate.

Top

Assumptions and Limitations

Top

Mathematical Description

The output Y can be expressed as:

Y=NOT(X1 AND X2 AND X3 AND X4)

Table 1. Truth Table

X4	X3	X2	X1	Y
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1

0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Top

Netlist Syntax

COUPL nand4 ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (tp_lh := @tp_lh , tp_hl := @tp_hl , x1 := @x1 , x2 := @x2 , x3 := @x3 , x4 := @x4) DST: SIM(Type:=SimVHDL) SRC: DB(File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
x1	Input Signal X1	bit	'0'
x2	Input Signal X2	bit	'0'
x3	Input Signal X3	bit	'0'
x4	Input Signal X4	bit	'0'
tp_lh	Rise Time Delay	real	0.0 [s]
tp_hl	Fall Time Delay	real	0.0 [s]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
y	Output Y	Output	bit

Top

Example

See Logic Gates Example: AND & NAND

Top

References

Two Input NOR Gate

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

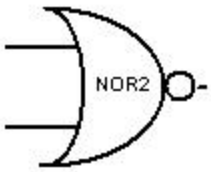

Figure 1. Component symbol

Description

The component represents a two input NOR gate.

Top

Assumptions and Limitations

Top

Mathematical Description

The output Y can be expressed as:

Y=NOT(X1 OR X2)

Table 1. Truth Table

X2	X1	Y
0	0	1
0	1	0
1	0	0
1	1	0

Top

Netlist Syntax

COUPL nor2 ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (tp_lh := @tp_lh , tp_hl := @tp_hl , x1 := @x1 , x2 := @x2) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
x1	Input Signal X1	bit	'0'
x2	Input Signal X2	bit	'0'
tp_lh	Rise Time Delay	real	0.0 [s]
tp_hl	Fall Time Delay	real	0.0 [s]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
y	Output Y	Output	bit

Top

Example

See Logic Gates Example: OR & NOR

Top

References

Two Input NOR Gate with One Inverted Input

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

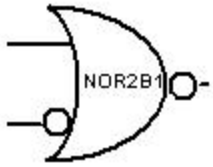

Figure 1. Component symbol

Description

The component represents a two input NOR gate with inverted input X2.

Top

Assumptions and Limitations

Top

Mathematical Description

The output Y can be expressed as:

Y=NOT(X1 OR X2')

Table 1. Truth Table

X2	X1	Y
0	0	0
0	1	0
1	0	1
1	1	0

Top

Netlist Syntax

COUPL nor2b1 ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (tp_lh := @tp_lh , tp_hl := @tp_hl , x1 := @x1 , x2 := @x2) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
x1	Input Signal X1	bit	'0'
x2	Input Signal X2	bit	'0'
tp_lh	Rise Time Delay	real	0.0 [s]
tp_hl	Fall Time Delay	real	[s]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
y	Output Y	Output	bit

Top

Example

See Logic Gates Example: OR & NOR

Top

References

Three Input NOR Gate

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

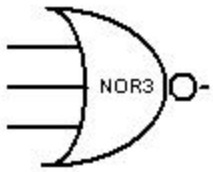

Figure 1. Component symbol

Description

The component represents a three input NOR gate.

Top

Assumptions and Limitations

Top

Mathematical Description

The output Y can be expressed as:

Y=NOT(X1 OR X2 OR X3)

Table 1. Truth Table

X3	X2	X1	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0

1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Top

Netlist Syntax

COUPL nor3 ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (tp_lh := @tp_lh , tp_hl := @tp_hl , x1 := @x1 , x2 := @x2 , x3 := @x3) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
x1	Input Signal X1	bit	'0'
x2	Input Signal X2	bit	'0'
x3	Input Signal X3	bit	'0'
tp_lh	Rise Time Delay	real	0.0 [s]
tp_hl	Fall Time Delay	real	0.0 [s]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
y	Output Y	Output	bit

Top

Example

See Logic Gates Example: OR & NOR

Top

References

Four Input NOR Gate

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

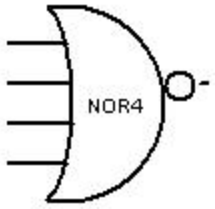

Figure 1. Component symbol

Description

The component represents a four input NOR gate.

Top

Assumptions and Limitations

Top

Mathematical Description

The output Y can be expressed as:

Y=NOT(X1 OR X2 OR X3 OR X4)

Table 1. Truth Table

X4	X3	X2	X1	Y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0

0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Top

Netlist Syntax

COUPL nor4 ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (tp_lh := @tp_lh , tp_hl := @tp_hl , x1 := @x1 , x2 := @x2 , x3 := @x3 , x4 := @x4) DST: SIM(Type:=SimVHDL) SRC: DB (File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
x1	Input Signal X1	bit	'0'
x2	Input Signal X2	bit	'0'
x3	Input Signal X3	bit	'0'
x4	Input Signal X4	bit	'0'
tp_lh	Rise Time Delay	real	0.0 [s]
tp_hl	Fall Time Delay	real	0.0 [s]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
y	Output Y	Output	bit

Top

Example

See Logic Gates Example: OR & NOR

Top

References

Two Input OR Gate

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

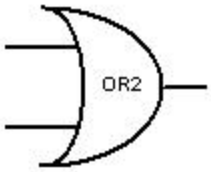

Figure 1. Component symbol

Description

The component represents a two input OR gate.

Top

Assumptions and Limitations

Top

Mathematical Description

The output Y can be expressed as:

Y =X1 OR X2

Table 1. Truth Table

X2	X1	Y
0	0	0
0	1	1
1	0	1
1	1	1

Top

Netlist Syntax

COUPL or2 ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (tp_lh := @tp_lh , tp_hl := @tp_hl , x1 := @x1 , x2 := @x2) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
x1	Input Signal X1	bit	'0'
x2	Input Signal X2	bit	'0'
tp_lh	Rise Time Delay	real	0.0 [s]
tp_hl	Fall Time Delay	real	0.0 [s]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
y	Output Y	Output	bit

Top

Example

See Logic Gates Example: OR & NOR

Top

References

Two Input OR Gate with One Inverted Input

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

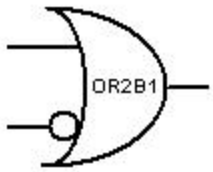

Figure 1. Component symbol

Description

The component represents a two input OR gate with inverted input X2.

Top

Assumptions and Limitations

Top

Mathematical Description

The output Y can be expressed as:

Y=X1 OR X2'

Table 1. Truth Table

X2	X1	Y
0	0	1
0	1	1
1	0	0
1	1	1

Top

Netlist Syntax

COUPL or2b1 ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (tp_lh := @tp_lh , tp_hl := @tp_hl , x1 := @x1 , x2 := @x2) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
x1	Input Signal X1	bit	'0'
x2	Input Signal X2	bit	'0'
tp_lh	Rise Time Delay	real	0.0 [s]
tp_hl	Fall Time Delay	real	0.0 [s]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
y	Output Y	Output	bit

Top

Example

See Logic Gates Example: OR & NOR

Top

References

Three Input OR Gate

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

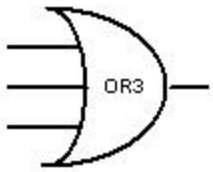

Figure 1. Component symbol

Description

The component represents a three input OR gate.

Top

Assumptions and Limitations

Top

Mathematical Description

The output Y can be expressed as:

Y=X1 OR X2 OR X3

Table 1. Truth Table

X3	X2	X1	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1

1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Top

Netlist Syntax

COUPL or3 ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (tp_lh := @tp_lh , tp_hl := @tp_hl , x1 := @x1 , x2 := @x2 , x3 := @x3) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
x1	Input Signal X1	bit	'0'
x2	Input Signal X2	bit	'0'
x3	Input Signal X3	bit	'0'
tp_lh	Rise Time Delay	real	0.0 [s]
tp_hl	Fall Time Delay	real	0.0 [s]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
y	Output Y	Output	bit

Top

Example

See Logic Gates Example: OR & NOR

Top

References

Four Input OR Gate

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

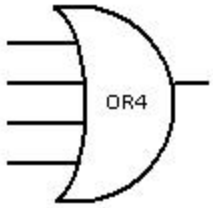

Figure 1. Component symbol

Description

The component represents a four input OR gate.

Top

Assumptions and Limitations

Top

Mathematical Description

The output Y can be expressed as:

Y=X1 OR X2 OR X3 OR X4

Table 1. Truth Table

X4	X3	X2	X1	Y
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1

0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Top

Netlist Syntax

COUPL or4 ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (tp_lh := @tp_lh , tp_hl := @tp_hl , x1 := @x1 , x2 := @x2 , x3 := @x3 , x4 := @x4) DST: SIM(Type:=SimVHDL) SRC: DB (File:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
x1	Input Signal X1	bit	'0'
x2	Input Signal X2	bit	'0'
x3	Input Signal X3	bit	'0'
x4	Input Signal X4	bit	'0'
tp_lh	Rise Time Delay	real	0.0[s]
tp_hl	Fall Time Delay	real	0.0[s]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
y	Output Y	Output	bit

Top

Example

See Logic Gates Example: OR & NOR

Top

References

Two Input XNOR Gate

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

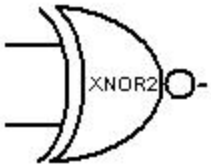

Figure 1. Component symbol

Description

The component represents a two input XNOR gate.

Top

Assumptions and Limitations

Top

Mathematical Description

The output Y can be expressed as:

Y=NOT(X1 XOR X2)

Table 1. Truth Table

X2	X1	Y
0	0	1
0	1	0
1	0	0
1	1	1

Top

Netlist Syntax

COUPL xnor2 ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (tp_lh := @tp_lh , tp_hl := @tp_hl , x1 := @x1 , x2 := @x2) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
x1	Input Signal X1	bit	'0'
x2	Input Signal X2	bit	'0'
tp_lh	Rise Time Delay	real	0.0 [s]
tp_hl	Fall Time Delay	real	0.0 [s]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
y	Output Y	Output	bit

Top

Example

See Logic Gates Example: XOR & XNOR & INV

Top

References

Three Input XNOR Gate

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

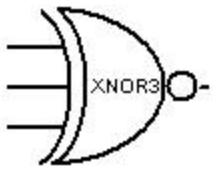

Figure 1. Component symbol

Description

The component represents a three input XNOR gate.

Top

Assumptions and Limitations

Top

Mathematical Description

The output Y can be expressed as:

Y=NOT(X1 XOR X2 XOR X3)

Table 1. Truth Table

X3	X2	X1	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1

1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Top

Netlist Syntax

COUPL xnor3 ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (tp_lh := @tp_lh , tp_hl := @tp_hl , x1 := @x1 , x2 := @x2 , x3 := @x3) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
x1	Input Signal X1	bit	'0'
x2	Input Signal X2	bit	'0'
x3	Input Signal X3	bit	'0'
tp_lh	Rise Time Delay	real	0.0 [s]
tp_hl	Fall Time Delay	real	0.0 [s]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
Y	Output Y	Output	bit

Top

Example

See Logic Gates Example: XOR & XNOR & INV

Top

References

Two Input XOR Gate

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

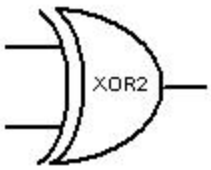

Figure 1. Component symbol

Description

The component represents a two input XOR gate.

Top

Assumptions and Limitations

Top

Mathematical Description

The output Y can be expressed as:

Y=X1 XOR X2

Table 1. Truth Table

X2	X1	Y
0	0	0
0	1	1
1	0	1
1	1	0

Top

Netlist Syntax

COUPL xor2 ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (tp_lh := @tp_lh , tp_hl := @tp_hl , x1 := @x1 , x2 := @x2) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
x1	Input Signal X1	bit	'0'
x2	Input Signal X2	bit	'0'
tp_lh	Rise Time Delay	real	0.0 [s]
tp_hl	Fall Time Delay	real	0.0 [s]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
y	Output Y	Output	bit

Top

Example

See Logic Gates Example: XOR & XNOR & INV

Top

References

Three Input XOR Gate

Library: Digital	Modeling Language: VHDL	Version Number: Twin Builder 2025.2

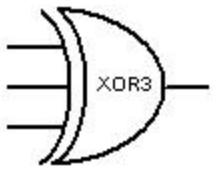

Figure 1. Component symbol

Description

The component represents a three input XOR gate.

Top

Assumptions and Limitations

Top

Mathematical Description

The output Y can be expressed as:

Y=X1 XOR X2 XOR X3

Table 1. Truth Table

X3	X2	X1	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0

1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Top

Netlist Syntax

COUPL xor3 ?InstanceName(@InstanceName):(@(Refbase)@(ID)) (tp_lh := @tp_lh , tp_hl := @tp_hl , x1 := @x1 , x2 := @x2 , x3 := @x3) DST: SIM(Type:=SimVHDL) SRC: DB(File:-:=@ModelLibraryName, Lang:=VHDLA, Lvl:=\\"@Architecture\\"); ;

Top

Parameters

Table 2

Name	Description	Data Type	Default Value [Unit]
x1	Input Signal X1	bit	'0'
x2	Input Signal X2	bit	'0'
x3	Input Signal X3	bit	'0'
tp_lh	Rise Time Delay	real	0.0 [s]
tp_hl	Fall Time Delay	real	0.0 [s]

Top

Input/Output Quantities

Table 3

Name	Description [Unit]	Direction	Data Type
y	Output Y	Output	bit

Top

Example

See Logic Gates Example: XOR & XNOR & INV

Top

References

Logic Gates Example: AND & NAND

This example illustrates the use of the AND and NAND Logic Gates.

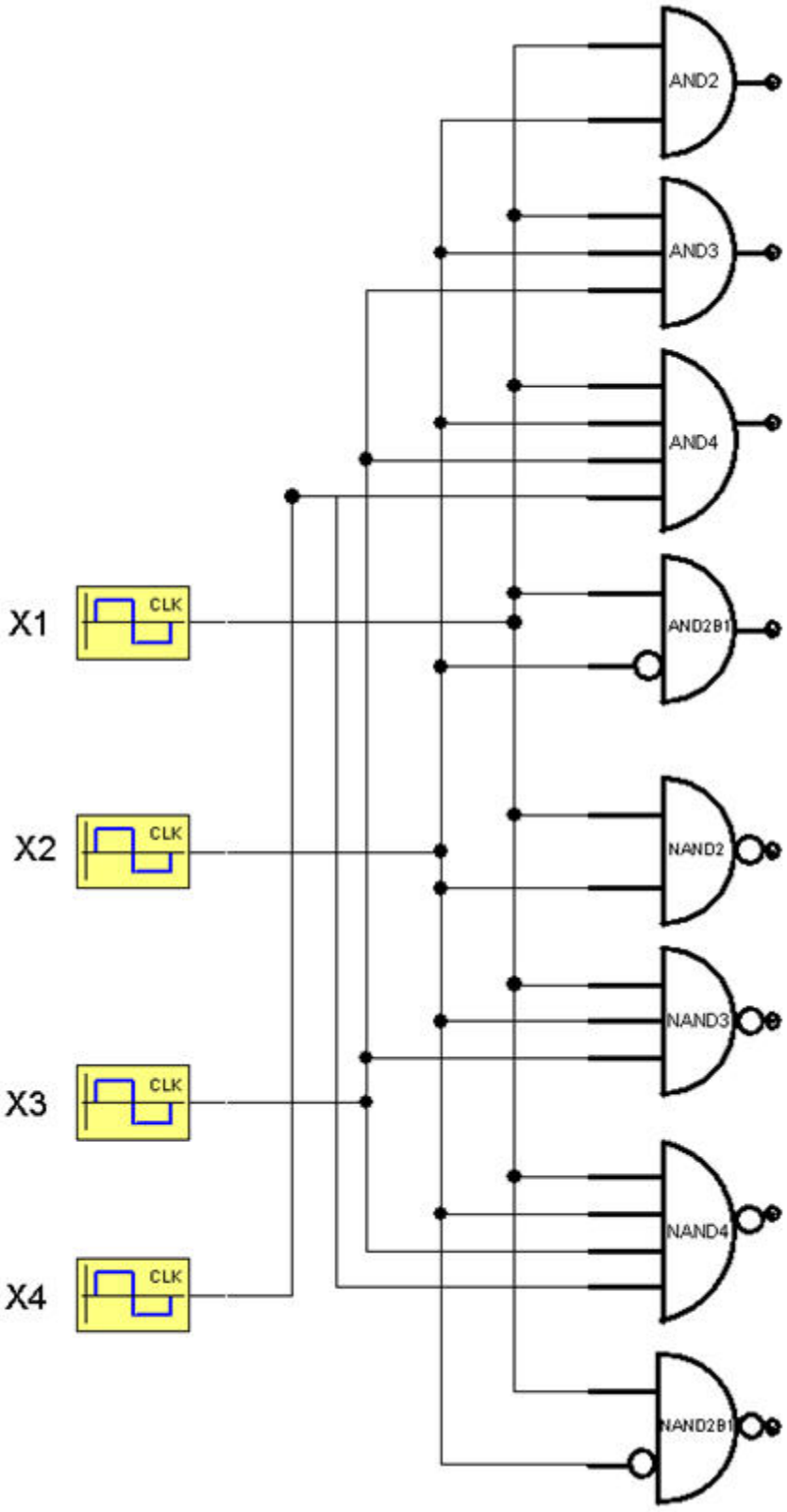

Figure 1. Application example of the AND and NAND Logic Gates

Table 1. System Parameters

Component	Parameter	Value [unit]
Clock Source X1	ped	2m [S]
	del	0 [S]
	periodical	1.0
Clock Source X2	ped	4m [S]
	del	0 [S]
	periodical	1.0
Clock Source X3	ped	8m [S]
	del	0 [S]
	periodical	1.0
Clock Source X4	ped	16m [S]
	del	0 [S]
	periodical	1.0
2-input AND Gate and21	tp_lh	0.0 [S]
	tp_hl	0.0 [S]
	x1	X1.val
	x2	X2.val
3-input AND Gate and31	tp_lh	0.0 [S]
	tp_hl	0.0 [S]

	x1	X1.val
	x2	X2.val
	x3	X3.val
4-input AND Gate and41	tp_lh	0.0 [S]
	tp_hl	0.0 [S]
	x1	X1.val
	x2	X2.val
	x3	X3.val
	x4	X4.val
2-input AND Gate with inverted input and2b11	tp_lh	0.0 [S]
	tp_hl	0.0 [S]
	x1	X1.val
	x2	X2.val
2-input NAND Gate nand21	tp_lh	0.0 [S]
	tp_hl	0.0 [S]
	x1	X1.val
	x2	X2.val
3-input NAND Gate nand31	tp_lh	0.0 [S]
	tp_hl	0.0 [S]
	x1	X1.val
	x2	X2.val

	x3	X3.val
4-input NAND Gate nand41	tp_lh	0.0 [S]
	tp_hl	0.0 [S]
	x1	X1.val
	x2	X2.val
	x3	X3.val
	x4	X4.val
2-input NAND Gate with inverted input nand2b11	tp_lh	0.0 [S]
	tp_hl	0.0 [S]
	x1	X1.val
	x2	X2.val

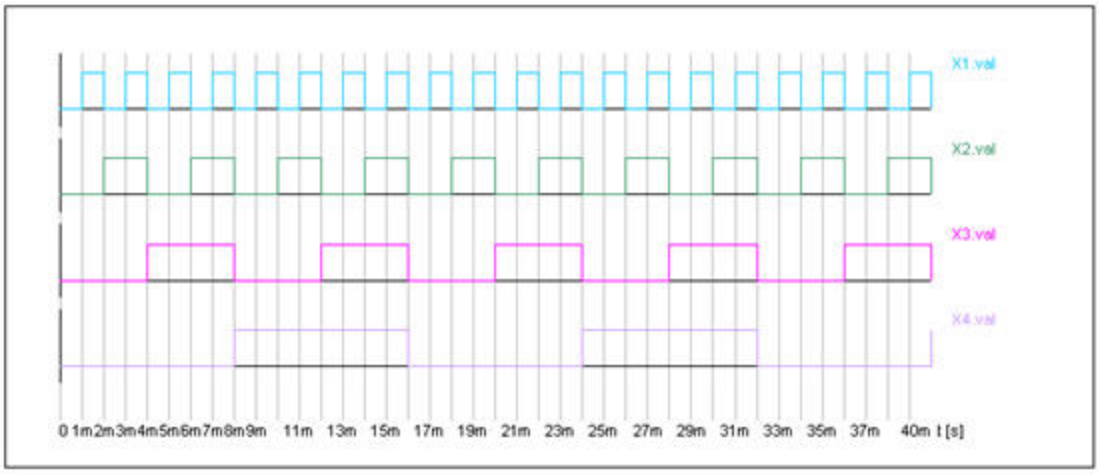

Figure 2. Simulation results-input to logic gates.

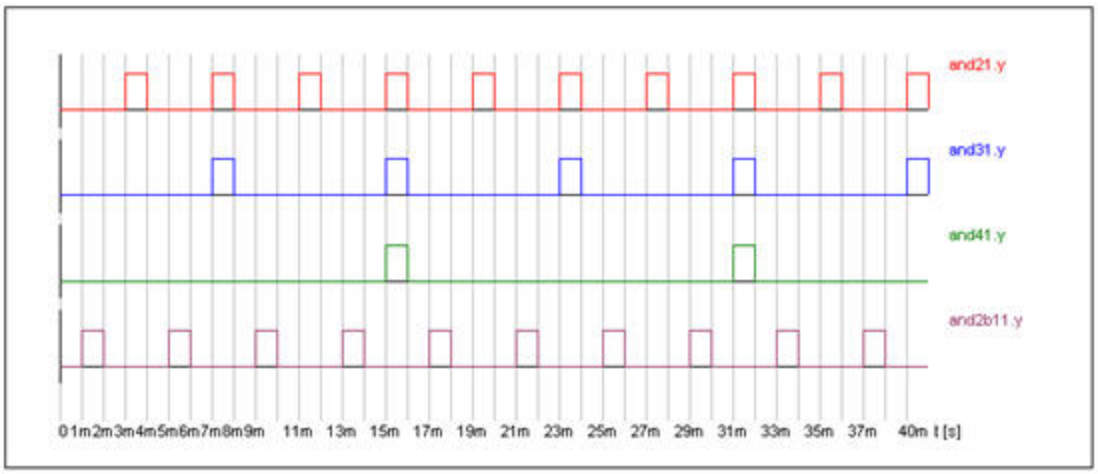

Figure 3. Simulation results-output of AND gate logic.

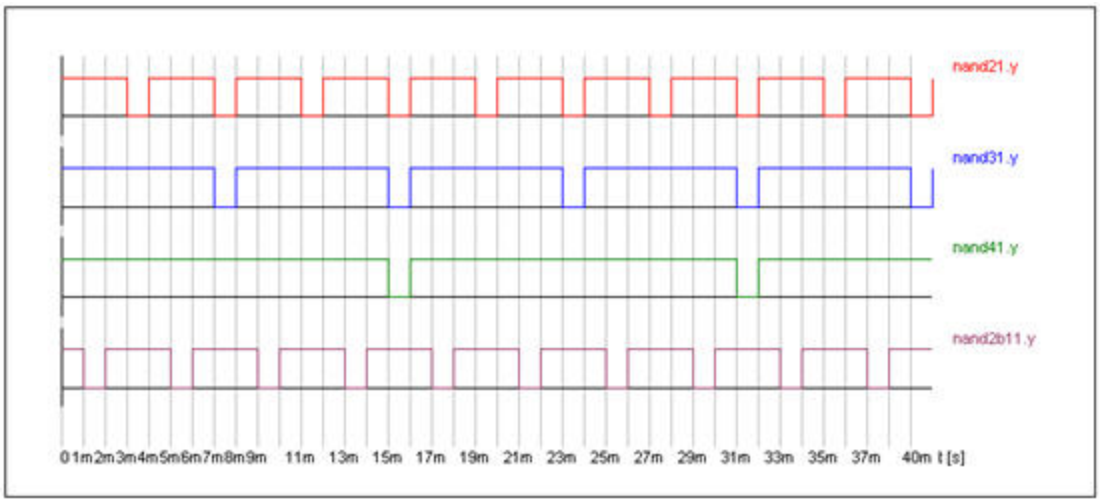

Figure 4. Simulation results-output of NAND gate logic.

Logic Gates Example: OR & NOR

This example illustrates the use of the AND and NAND Logic Gates.

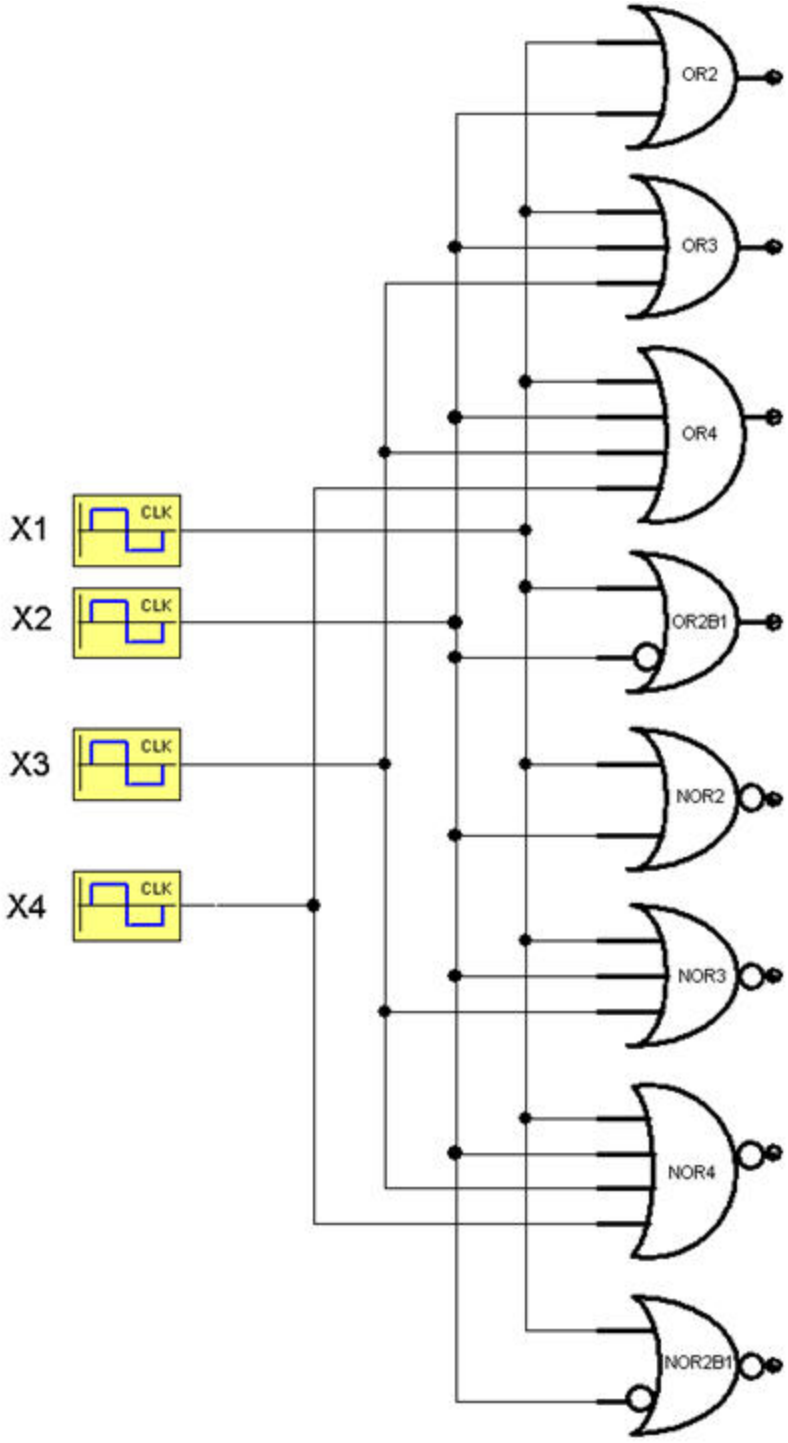

Figure 1. Application example of the OR and NOR Logic Gates

Table 1. System Parameters

Component	Parameter	Value [unit]
Clock Source X1	ped	2m [S]
	del	0 [S]
	periodical	1.0
Clock Source X2	ped	4m [S]
	del	0 [S]
	periodical	1.0
Clock Source X3	ped	8m [S]
	del	0 [S]
	periodical	1.0
Clock Source X4	ped	16m [S]
	del	0 [S]
	periodical	1.0
2-input OR Gate or21	tp_lh	0.0 [S]
	tp_hl	0.0 [S]
	x1	X1.val
	x2	X2.val
3-input OR Gate or31	tp_lh	0.0 [S]
	tp_hl	0.0 [S]

	x1	X1.val
	x2	X2.val
	x3	X3.val
4-input OR Gate or41	tp_lh	0.0 [S]
	tp_hl	0.0 [S]
	x1	X1.val
	x2	X2.val
	x3	X3.val
	x4	X4.val
2-input OR Gate with inverted input or2b11	tp_lh	0.0 [S]
	tp_hl	0.0 [S]
	x1	X1.val
	x2	X2.val
2-input NOR Gate nor21	tp_lh	0.0 [S]
	tp_hl	0.0 [S]
	x1	X1.val
	x2	X2.val
3-input NOR Gate nor31	tp_lh	0.0 [S]
	tp_hl	0.0 [S]
	x1	X1.val
	x2	X2.val

	x3	X3.val
4-input NOR Gate nor41	tp_lh	0.0 [S]
	tp_hl	0.0 [S]
	x1	X1.val
	x2	X2.val
	x3	X3.val
	x4	X4.val
2-input NOR Gate with inverted input nor2b11	tp_lh	0.0 [S]
	tp_hl	0.0 [S]
	x1	X1.val
	x2	X2.val

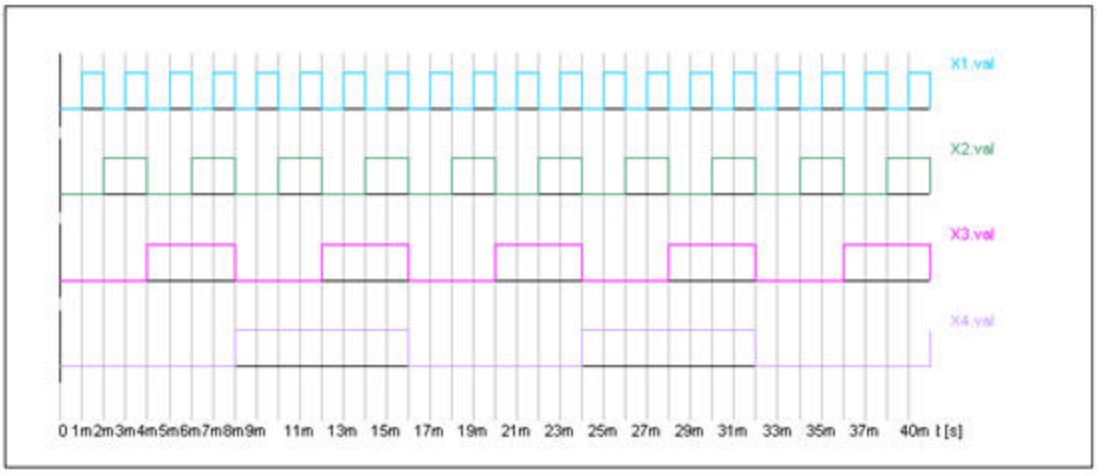

Figure 2. Simulation results-input to logic gates.

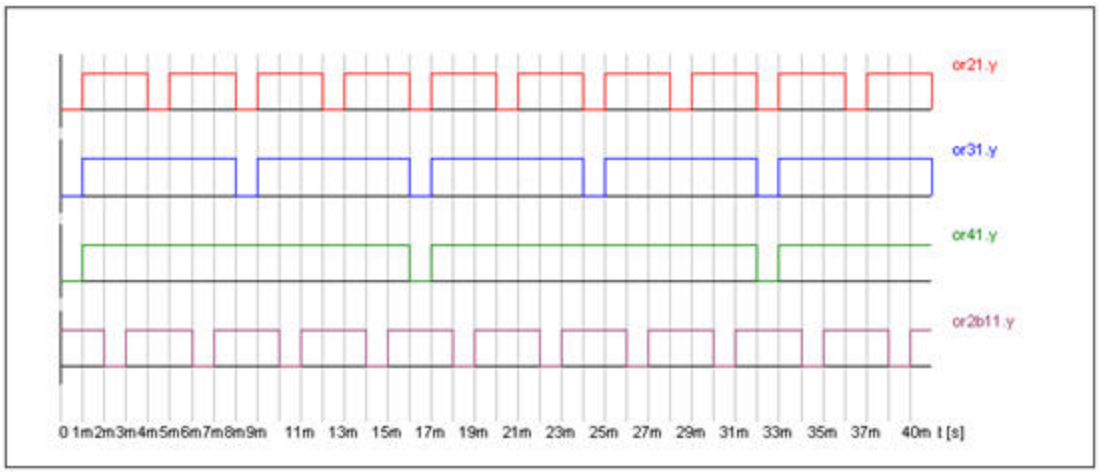

Figure 3. Simulation results-output of OR gate logic.

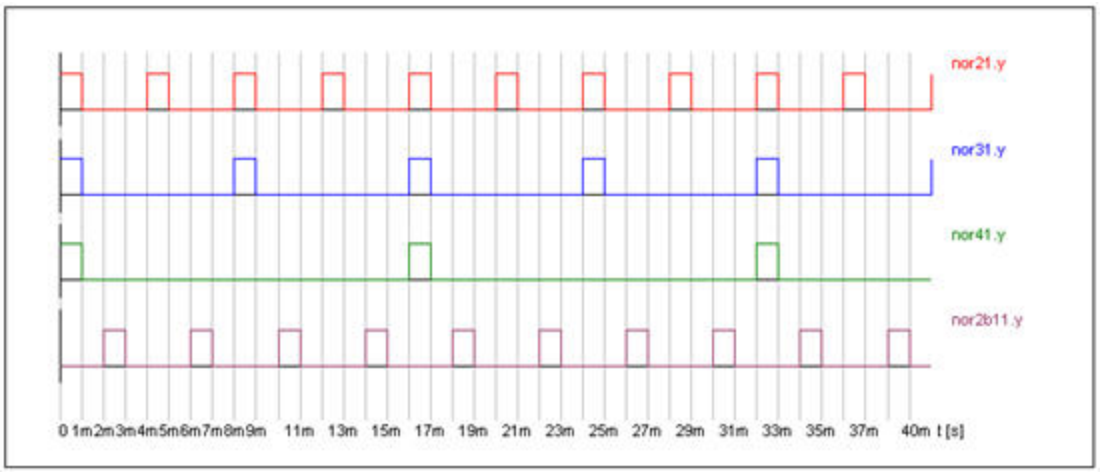

Figure 4. Simulation results-output of NOR gate logic.

Logic Gates Example: XOR & XNOR & INV

This example illustrates the use of the XOR and XNOR Logic Gates and the INV gate.

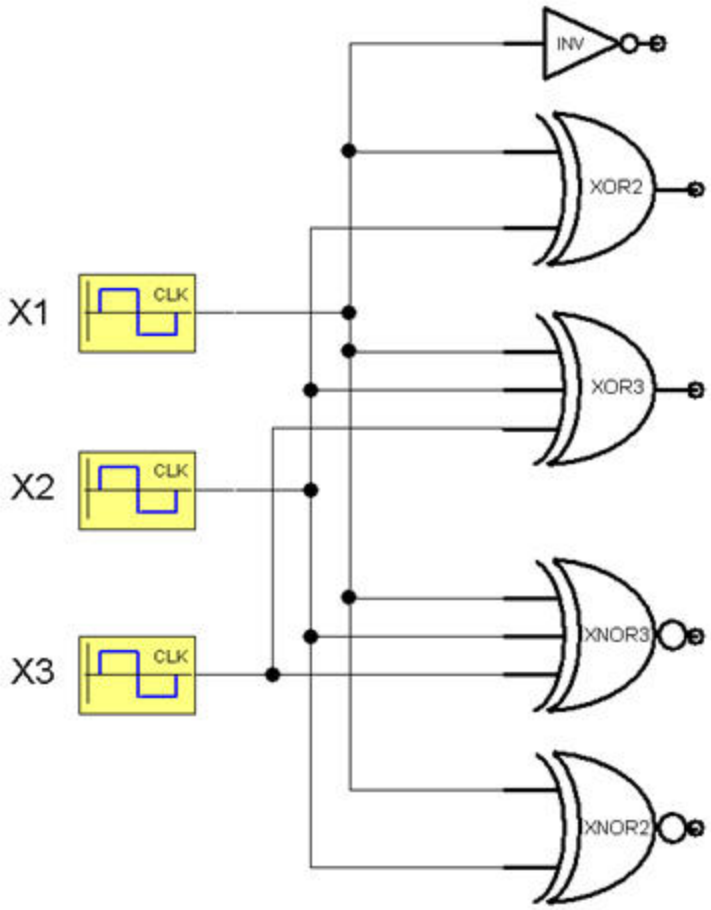

Figure 1. Application example of the XOR and XNOR Logic Gates and the INV gate

Table 1. System Parameters

Component	Parameter	Value [unit]
Clock Source X1	ped	4m [S]
	del	0 [S]
	periodical	1.0
Clock Source X2	ped	8m [S]

	del	0 [S]
	periodical	1.0
Clock Source X3	ped	16m [S]
	del	0 [S]
	periodical	1.0
2-input XOR Gate xor21	tp_lh	0.0 [S]
	tp_hl	0.0 [S]
	x1	X1.val
	x2	X2.val
3-input XOR Gate xor31	tp_lh	0.0 [S]
	tp_hl	0.0 [S]
	x1	X1.val
	x2	X2.val
	x3	X3.val
2-input XNOR Gate xnor21	tp_lh	0.0 [S]
	tp_hl	0.0 [S]
	x1	X1.val
	x2	X2.val
3-input XNOR Gate xnor31	tp_lh	0.0 [S]
	tp_hl	0.0 [S]

	x1	X1.val
	x2	X2.val
	x3	X3.val
Inverter Gate inv1	tp_lh	0.0 [S]
	tp_hl	0.0 [S]
	x1	X1.val

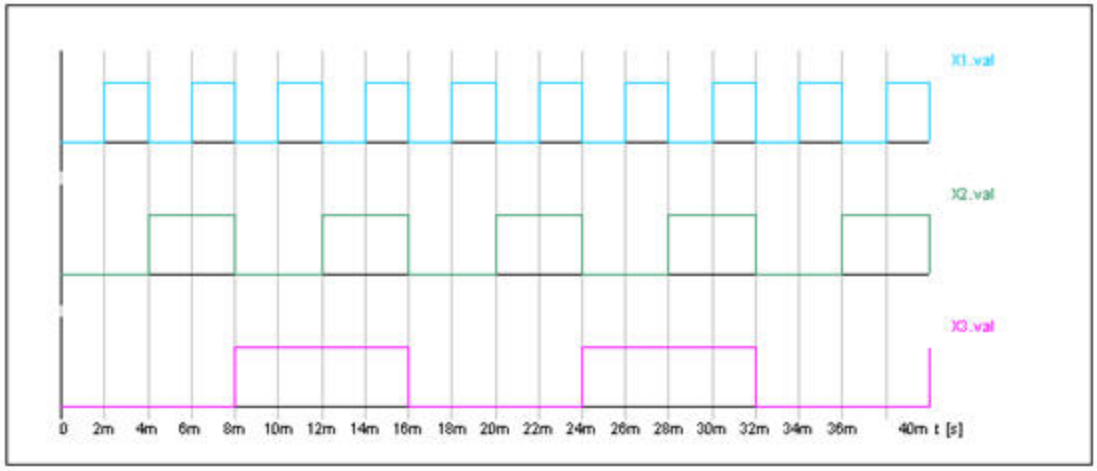

Figure 2. Simulation results-input to logic gates.

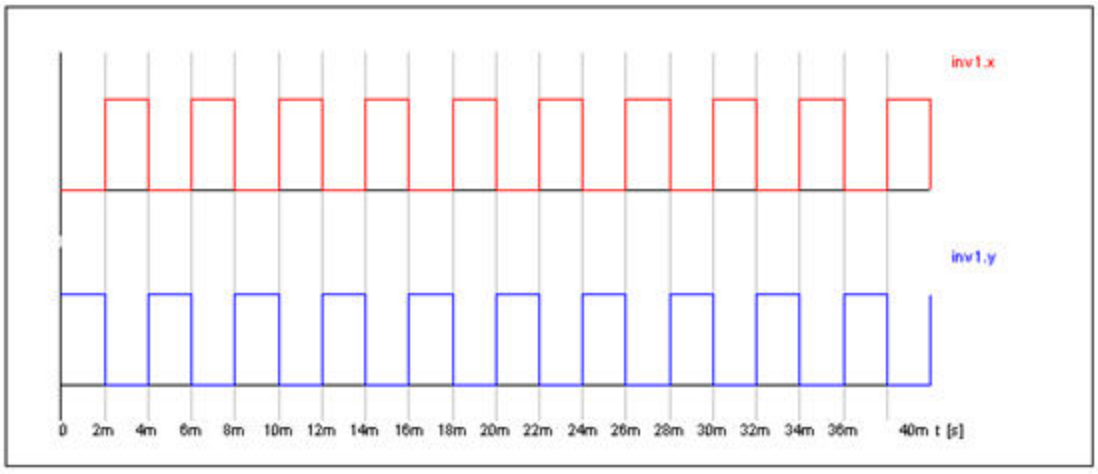

Figure 3. Simulation results-output of INV gate logic.

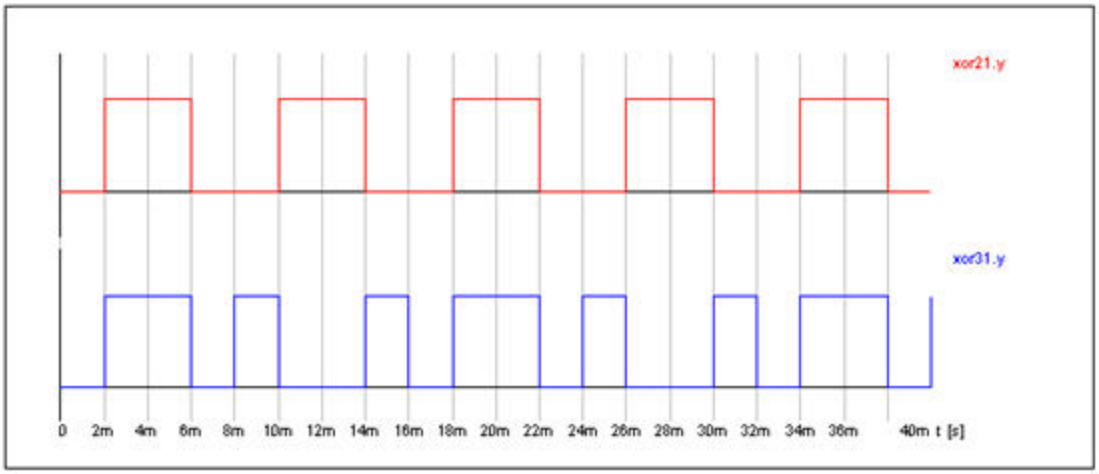

Figure 4. Simulation results-output of XOR gate logic.

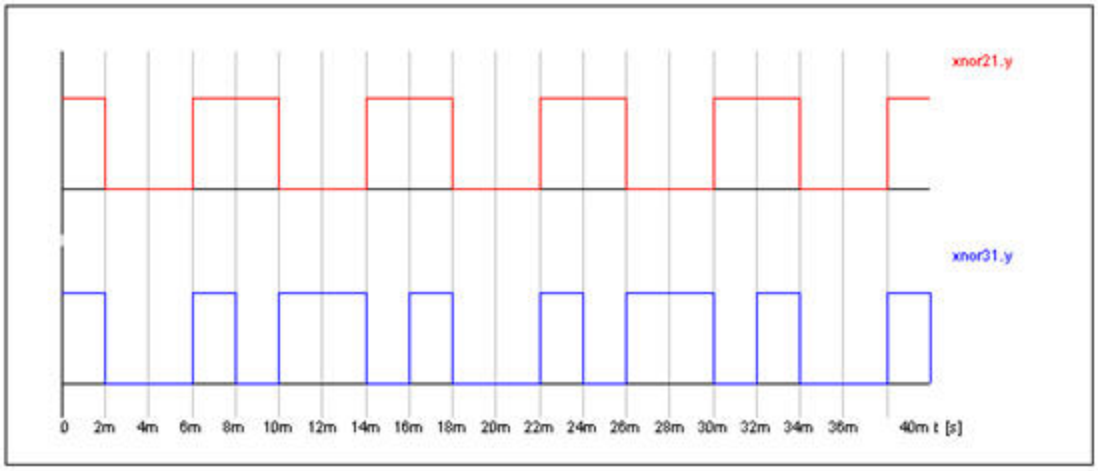

Figure 5. Simulation results-output of XNOR gate logic.

Index