
Functional VHDL-AMS Model Description

ATA6662/ATA6662C/ATA6663/ATA6664

Introduction

This documentation provides an overview of the functional VHDL-AMS model of the Atmel® LIN transceivers ATA6662, ATA6662C, ATA6663, and ATA6664. The LIN transceivers have nearly the same behavior, the slight differences are described in an extra section.

Unfortunately, some parts of the VHDL-AMS language standard are not implemented yet in all simulators so that simulator-dependent differences exist in the model (see [Section 3. “Using the Model” on page 10](#)).

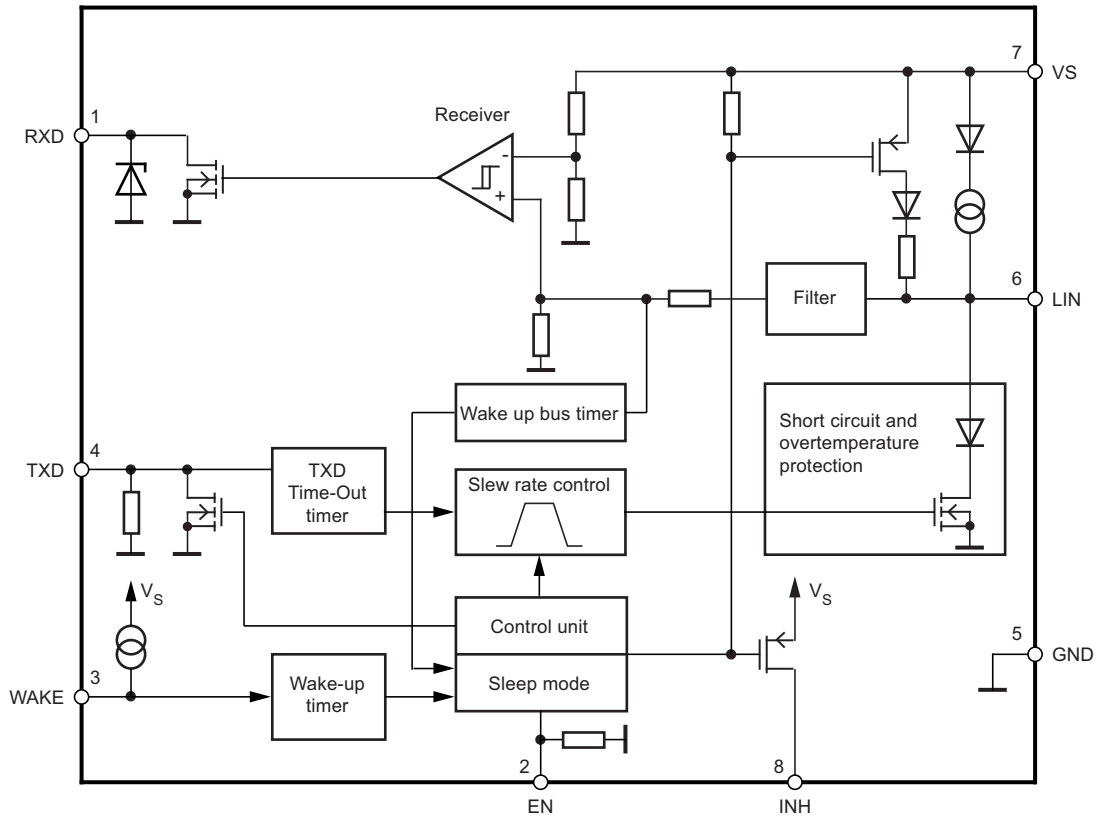
1. Functional Description

1.1 Overview

The Atmel® LIN transceiver ATA666x (see [Figure 1-1](#)) serves as an interface between the LIN protocol handler and the physical layer of the LIN specification 2.1. It includes a logic part (control unit, slew rate control, wake-up timer for bus and local wake, TXD time-out timer) as well as an analog part consisting of the LIN driver (at pin LIN), the LIN pull-up circuit (between pin VS and LIN), the receiver circuit (comparator and RXD driver with open drain at pin RXD), the TXD pull-down transistor (at pin TXD), the pull-up current source (at pin WAKE), and the inhibit switch (transistor at pin INH).

For further details please see the corresponding datasheet.

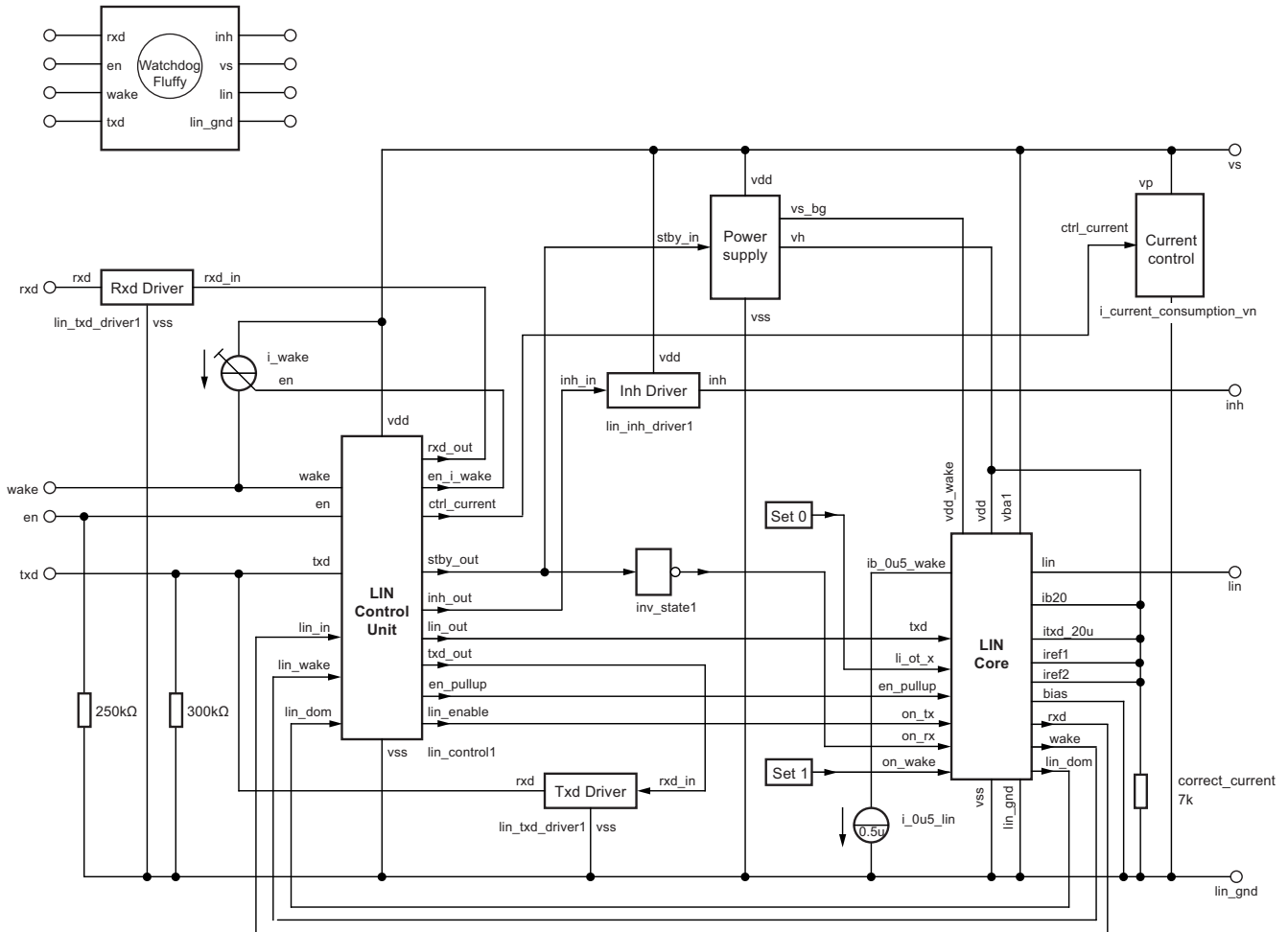
Figure 1-1. Functional Block Diagram of the LIN Transceiver



1.2 Behavioral Model

The model represents all functionality of the LIN transceiver. The behavior at LIN is modeled in detail while the behavior at the other pins is described in more ideal terms. The complete control is implemented digitally to allow faster simulation.

Figure 1-2. Schematic of the LIN Transceiver



The LIN control unit is modeled as a finite state machine to control the different states of the LIN transceiver (see Section 3.10 of the datasheet). The current state is represented by an internal digital state *lin_state*. Its value could be one the following:

- 0 ==> unpowered mode
- 1 ==> pre-normal/fail-safe mode
- 2 ==> normal mode
- 3 ==> sleep mode

Figure 1-3. States of the LIN Transceiver

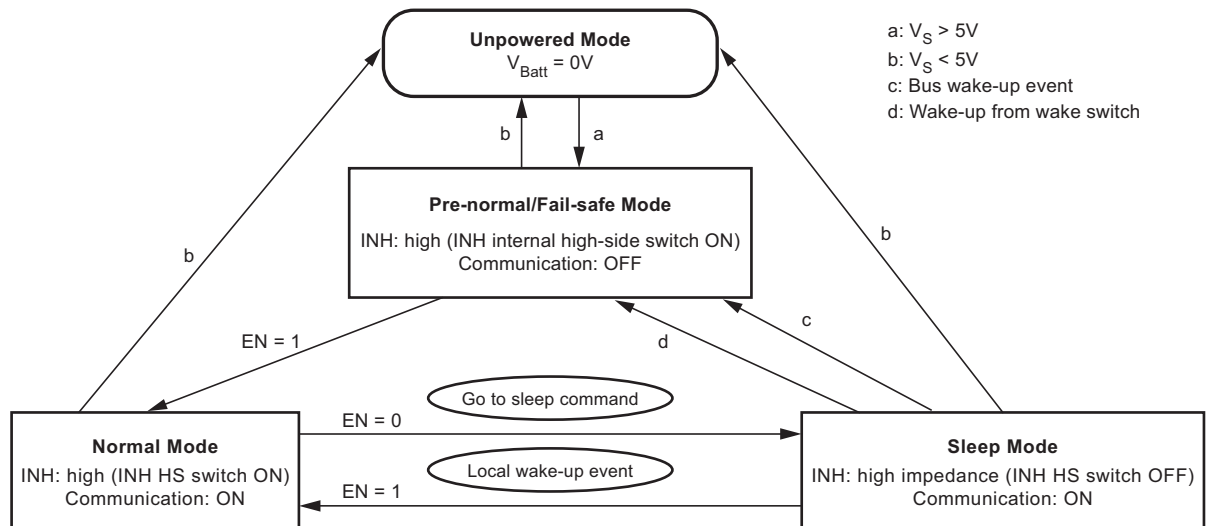
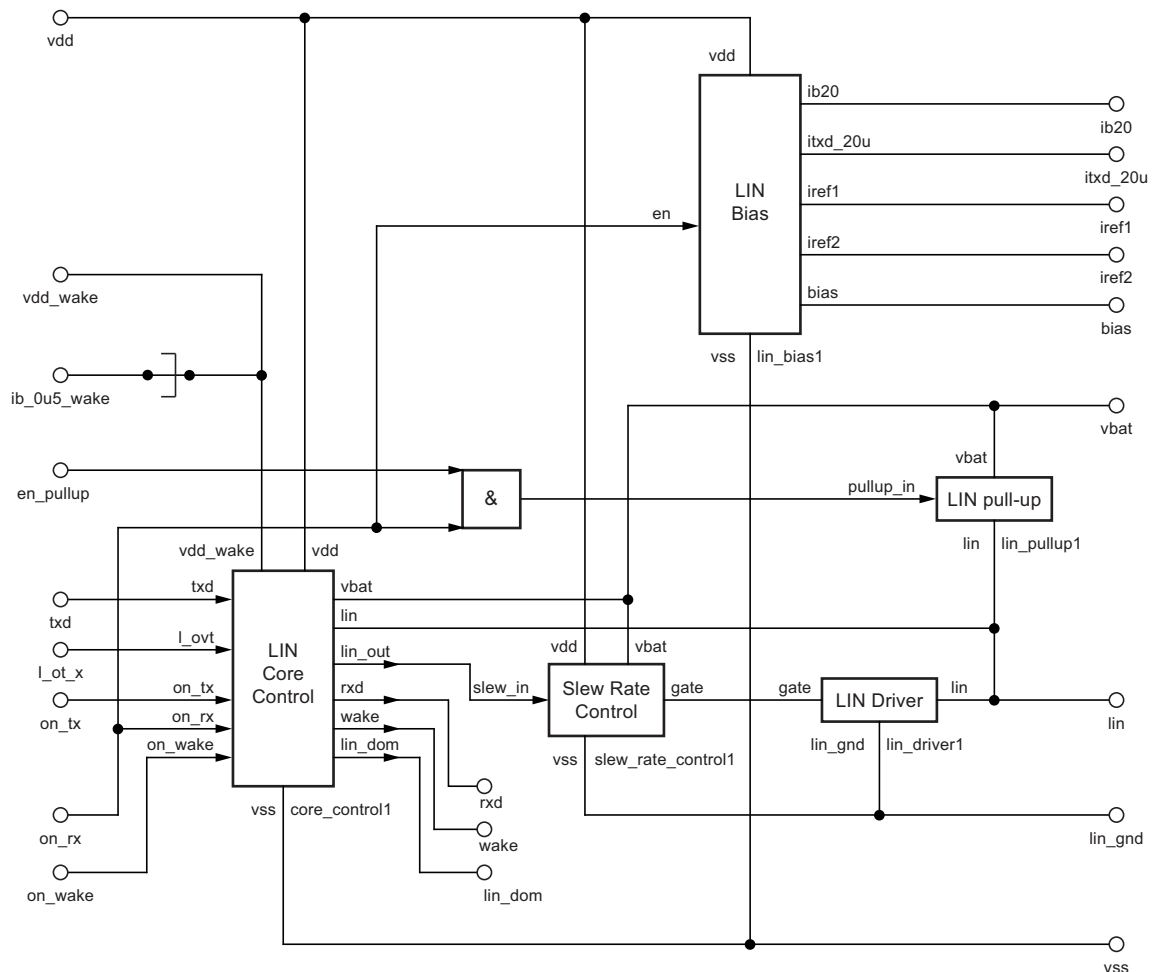


Figure 1-4. Schematic of the LIN Core



The LIN core also has a control block (LIN core control) because there are control pins within the LIN core (see [Figure 1-4 on page 4](#)). The communication between the state machine and the LIN core control is realized by control signals such as those in the real chip.

In normal mode, a falling edge at pin TXD leads to a falling edge at pin LIN. The signal *lin_out* of the LIN core control provokes a ramp in the slew rate control with defined ramping on *gate*. The voltage at the pin gate controls the output HVMOS transistor within the LIN driver.

The analog behavior of the other pins is modeled by switched resistors and switched current sources respectively. Non-ideal behavior is a result of the external loads at the pins.

Signal transitions within the model are delayed if there is real-time latency. All latency times of the model are defined in the file *lin_parameters.vhd* for three characteristic cases.

In addition to the latency times, model-specific parameters are also defined. One of the three parameter sets can be selected by setting the model parameter *parameter_set* to one of the following items:

```
typ ==> typical parameter set
min ==> lower bound of parameters (best case)
max ==> upper bound of parameters (worst case)
```

The items have to be enclosed in apostrophes because *parameter_set* is a string type.

1.3 Watchdog

The LIN transceiver model additionally includes a watchdog monitoring the voltages at the outer pins (VS, INH, LIN, LIN_GND, RXD, TXD, WAKE, EN). It issues both warnings as well as error messages if the voltages cross certain thresholds.

The thresholds for warning messages are listed in [Table 1-1](#).

Table 1-1. Thresholds for Warning Messages

Pin (Direction)	Parameters		Unit
	Minimum	Maximum	
VS (input)	5	18	V
INH (output)	−0.3	+40	V
LIN (input/output)	−27	+40	V
RXD (output)	−0.3	+6	V
TXD (input / output)	−0.3	+6	V

The thresholds for error messages are listed in [Table 1-2](#). The simulation aborts if an error occurs.

Table 1-2. Thresholds for Error Messages

Pin (Direction)	Parameters		Unit
	Minimum	Maximum	
VS (input)	−0.3	+40	V
WAKE (input)	−1	+40	V
EN (input)	−0.3	+6	V

1.4 Functionality

The model has a parameter *lin_type* to use it as model of the Atmel® ATA6662, the Atmel ATA6663, or the Atmel ATA6664. One of the three types can be selected by setting the model parameter *lin_type* to one of the following items:

6662 ==> Atmel ATA6662 (nominal value in Atmel ATA6662 packages)
 6663 ==> Atmel ATA6663 (nominal value in Atmel ATA6663 packages)
 6664 ==> Atmel ATA6664 (nominal value in Atmel ATA6664 packages)

The items have to be enclosed in apostrophes because *lin_type* is a string type. If *lin_type* is set to another value than specified, the ATA6662 is used.

The models are nearly identical. The differences are listed In [Table 1-3](#).

Table 1-3. Differences of the Models

Spec. Point	Function	Atmel ATA6662			Atmel ATA6663			Atmel ATA6664		
		Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.
5.3	Switch-on resistance INH to VS	300Ω	560Ω	800Ω	10Ω	30Ω	50Ω	10Ω	30Ω	50Ω
9.5	TXD dominant time out timer tdom	6ms	9ms	20ms	27ms	55ms	70ms	No time out		
	Time-out on floating LIN				10ms	10ms	10ms	10ms	10ms	10ms
	Behavior	If TXD is low and afterwards EN switches to high, then the LIN switches to low			If TXD is low and afterwards EN switches to high, then the LIN does not switch to low.			If TXD is low and afterwards EN switches to high, then the LIN does not switch to low.		
	Behavior				The internal pull-up resistor is only enabled in normal mode (<i>lin_state</i> =2).			The internal pull-up resistor is only enabled in normal mode (<i>lin_state</i> =2).		
	LIN bus wake-up	Wake-up if LIN is below LIN wake level for <i>t_bus</i>			Wake-up if LIN is in dominant state for <i>t_bus</i> and this state is reached within <i>t_float_6663</i> after LIN is below LIN wake level			Wake-up if LIN is in dominant state for <i>t_bus</i> and this state is reached within <i>t_float_6663</i> after LIN is below LIN wake level		
	Current consumption	Realized by the internal resistors and current sources			Realized by the additional module <i>lin_current_control</i>			Realized by the additional module <i>lin_current_control</i>		

The model uses different parameters in 6663- and 6664-mode:

t_dom_6663 instead of *t_dom*

r_inh_6663 instead of *r_inh*

t_float_6663 for time-out if LIN is floating or hot-wired; ATA6662 has no equivalent

Figure 1-5. Different Behavior of Atmel ATA6662 and ATA6663

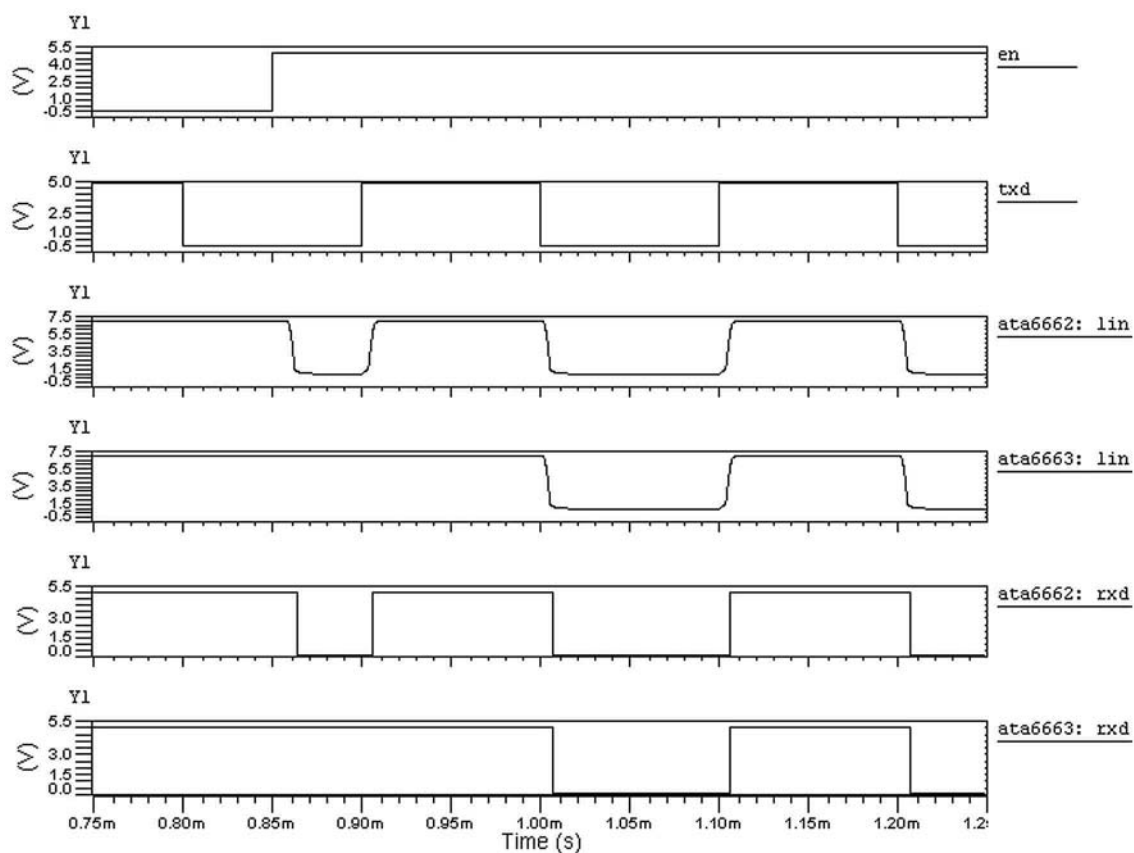


Figure 1-6. TXT Time-out of Atmel ATA6662, ATA6663, and ATA6664 (*lin_ata666x_tb_E.vhd*)

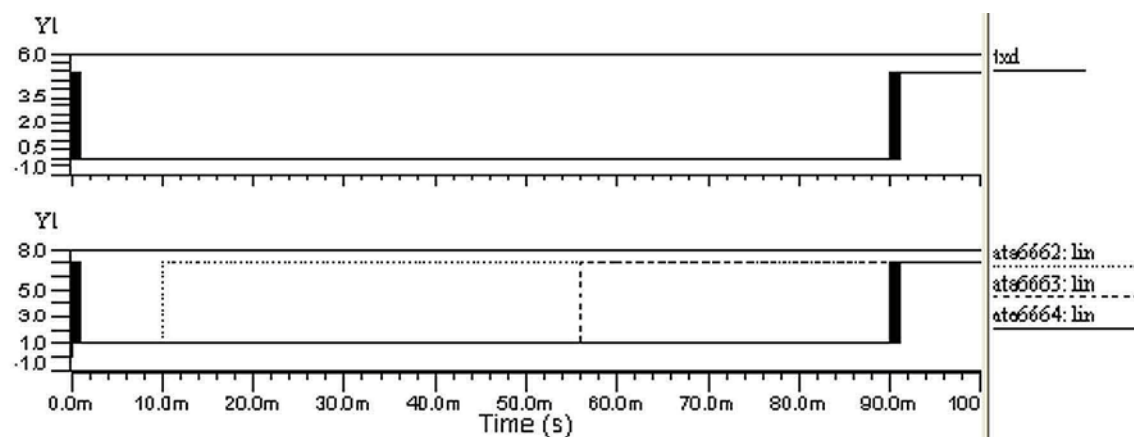
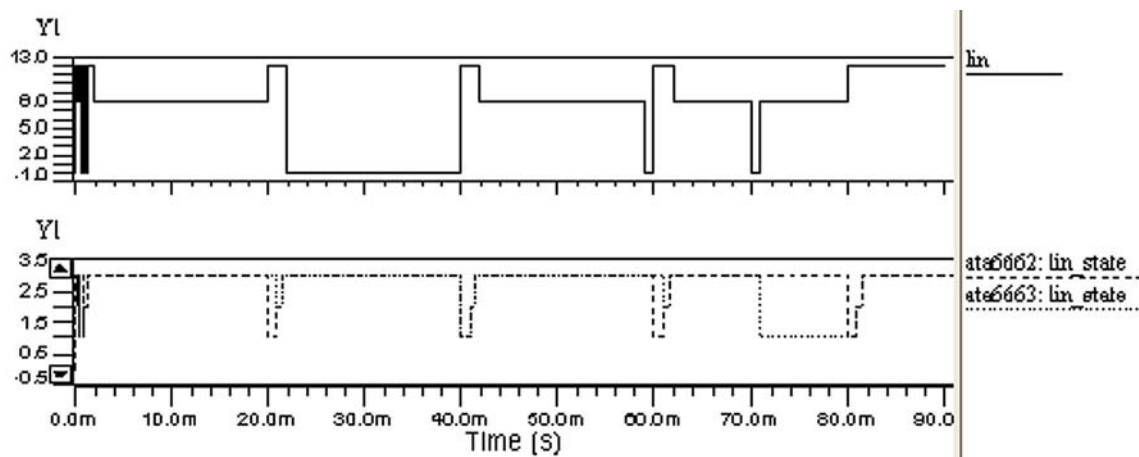


Figure 1-7. LIN Bus Wake-up Behavior of Atmel ATA6662 and ATA6663 (*lin_ata666x_tb_F.vhd*)



2. Interface Declaration

Table 2-1 provides an overview of the interfaces of the Atmel LIN transceiver ATA6662, including pin names and their descriptions.

Table 2-1. Interfaces of the Atmel LIN Transceiver

No.	Pin Name	Pin Type	Signal Description	Comment
1	RXD	Electrical	Receives data output	The output is an open drain.
2	EN	Electrical	Enables normal mode	When the input is open or low, the device is in sleep mode (provided that $V_S > 7V$).
3	WAKE	Electrical	High-voltage input for local wake-up request	The voltage threshold for a wake-up signal is 3V below the voltage at VS.
4	TXD	Electrical	Transmits data input	Active low output (strong pull-down) after a local wake-up request
5	LIN_GND	Electrical	Ground	
6	LIN	Electrical	LIN bus input/output	
7	VS	Electrical	Supply voltage	
8	INH	Electrical	Inhibit output for controlling an external voltage regulator	Active high output after a wake-up request

3. Using the Model

The model is available on the Atmel® website under the corresponding LIN transceiver.

3.1 Model Parameters

There are two parameters, one parameter *lin_type* to select one of the three LIN types (6662, 6663, 6664) and a second parameter *parameter_set* to select one of three pre-defined parameter sets (*typ*, *min*, *max*). The nominal parameter set *typ* is preselected. All parameter values (see [Section 5. “Modeling Parameters” on page 12](#)) are defined in the file *lin_parameters.vhd* and can be changed where appropriate.

3.2 Simulator Call

When simulating a test bench, no specific parameters need to be passed to the simulator. Thus, the respective test bench is called either by running a prepared batch for Questa ADMS (AdvanceMS), running a prepared batch for Virtuoso AMS Designer, or by opening a prepared project for SystemVision.

Because the Virtuoso AMS Designer has problems with the VHDL-AMS break mechanism, Atmel recommends using the native Verilog break mechanism by setting the parameter

`fastbreak = yes`

in each transient simulation. Otherwise the CPU time may rise by a factor of 4000.

3.3 Simulator-depending Differences

Some parts of the VHDL-AMS language standard are not implemented yet in all simulators.

SystemVision is a fast, native VHDL-AMS simulator. Correct handling of break and 'above is implemented, but there are numerical problems without break and 'above.

Virtuoso AMS Designer⁽¹⁾ is a native Verilog-AMS simulator. So VHDL-AMS is mapped to the solver as far as possible. In some cases it is necessary to

- remove *break* statements
- replace *x'above(y)* by *x > y*
- remove *'ramp()* and *'slew()*

Furthermore some statements are not implemented yet:

- comparing strings
- selecting values from a vector in equations
- non-scalar parameters
- impure functions

Unfortunately, SystemVision cannot handle all the work-arounds required for the Virtuoso AMS Designer. So there is a special package for the Virtuoso AMS Designer with a slightly different model of the LIN output driver *lin_lin_driver.vhd*.

Note: 1. The model doesn't work with the version 06.11-S012.

Questa ADMS (AdvanceMS) is a very stable, native VHDL-AMS simulator. The package contains the same VHDL-AMS models as the SystemVision package. But there are no problems using the AMS Designer package models.

4. Analysis Types

Table 4-1. Analysis Types

Analysis Type	Short Cut	Supported/Not Supported
DC Analysis	OP	Supported
DC Transfer Analysis	DC	Supported
Transient Analysis	TRANS	Supported
AC Analysis	AC	Not supported
Noise Analysis	NOISE	Not supported
Distortion Analysis	DISTO	Not supported
Sensitivity Analysis	SENS	Not supported
Monte Carlo Analysis	MC	Not supported

5. Modeling Parameters

The following parameters are defined in the file *lin_parameters.vhd*.

Table 5-1. Modelling Parameters

Parameter's Description	Symbol	Min.	Typ.	Max.	Unit
LIN Control Unit					
EN: threshold voltage for high-level input	v_enh	2	2	2	V
EN: threshold voltage for low-level input	v_enl	0.8	0.8	0.8	V
TXD: threshold voltage for high-level input	v_txdh	2	2	2	V
TXD: threshold voltage for low-level input	v_txdl	0.8	0.8	0.8	V
WAKE: max. voltage difference to VS for high-level input	add_wakeh	−1	−1	−1	V
WAKE: min. voltage difference to VS for low-level input	add_wakel	−3	−3	−3	V
VS: undervoltage threshold	v_min	4	4.6	5	V
Dominant time for wake-up via LIN bus	t_bus	30	90	150	μs
Time of low pulse for wake-up via pin WAKE	t_wake	7	20	50	μs
Minimum time of high level before low pulse at pin WAKE	t_wake_reset	6	6	6	μs
Time delay for mode change from pre-normal/fail-safe mode to normal mode via pin EN	t_norm	2	10	15	μs
Time delay for mode change from normal mode to sleep mode via pin EN	t_sleep	2	10	12	μs
TXD dominant time out timer (Atmel ATA6662)	t_dom	6	9	20	ms
TXD dominant time out timer (Atmel ATA6663)	t_dom_6663	27	55	70	ms
Power-up delay between $V_S \geq v_min$ until INH switches to high level	t_vs	50	100	200	μs
Time-out if LIN is floating or hot-wired (Atmel ATA6663/ATA6664)	t_float_6663	10	10	10	ms
LIN Core					
WAKE: propagation delay of falling edge at pin LIN if $V_S = 7.3V$	t_wake_pdf_73	1.3	1.5	1.7	μs
WAKE: propagation delay of falling edge at pin LIN if $V_S = 18V$	t_wake_pdf_18	1.9	2	2.2	μs
WAKE: propagation delay of rising edge at pin LIN if $V_S = 7.3V$	t_wake_pdr_73	1.3	1.5	1.7	μs
WAKE: propagation delay of rising edge at pin LIN if $V_S = 18V$	t_wake_pdr_18	1.9	2	2.2	μs
RXD: propagation delay of receiver for falling edge at pin LIN if $V_S = 7.3V$	t_rx_pdf_73	2.7	3.3	3.9	μs
RXD: propagation delay of receiver for falling edge at pin LIN if $V_S = 18V$	t_rx_pdf_18	2.2	2.7	3.1	μs
RXD: propagation delay of receiver for rising edge at pin LIN if $V_S = 7.3V$	t_rx_pdr_73	1	2.5	3.9	μs
RXD: propagation delay of receiver for rising edge at pin LIN if $V_S = 18V$	t_rx_pdr_18	0.9	2.4	3.9	μs
TXD: propagation delay of LIN driver to falling edge at pin LIN	t_trans_pdf	1.3	1.5	1.7	μs
TXD: propagation delay of LIN driver to rising edge at pin LIN	t_trans_pdr	2.4	3.5	4.5	μs
LIN: slope fall (load: $R = 1k\Omega$, $C = 1nF$)	t_slope_fall	5	6	7	μs
LIN: slope rise (load: $R = 1k\Omega$, $C = 1nF$)	t_slope_rise	8	10	12	μs
LIN: low-level threshold voltage $V_{th_dom} = V_S \times x_linl$, $x_linl = (V_{BUS_CNT} - 0.5 \times V_{BUS_HYS}) / [V]$	x_linl	0.461	0.45	0.4375	
LIN: high-level threshold voltage $V_{th_rec} = V_S \times x_linh$, $x_linh = (V_{BUS_CNT} + 0.5 \times V_{BUS_HYS}) / [V]$	x_linh	0.503	0.55	0.6125	
LIN: pull-up resistor to VS	r_pullup	20	40	60	kΩ
Pull-up Current Source at Pin WAKE					
Wake current	i_wake	0.5	10	30	μA
INH Output Pin					
Switch on resistance to VS (Atmel ATA6662)	r_inh	300	560	800	Ω
Switch on resistance to VS (Atmel ATA6663/ATA6664)	r_inh_6663	10	30	50	Ω

Figure 5-1. Waveform Diagrams

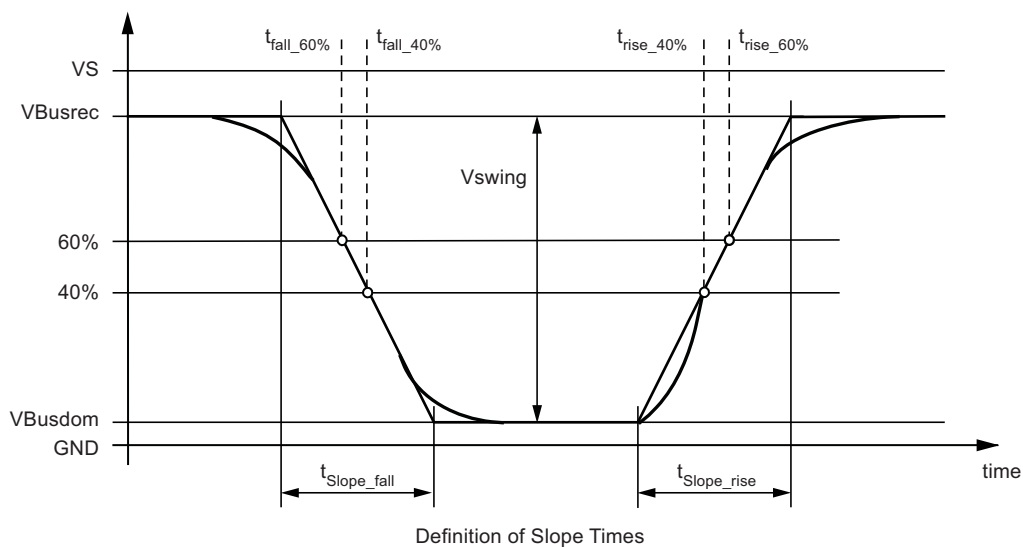
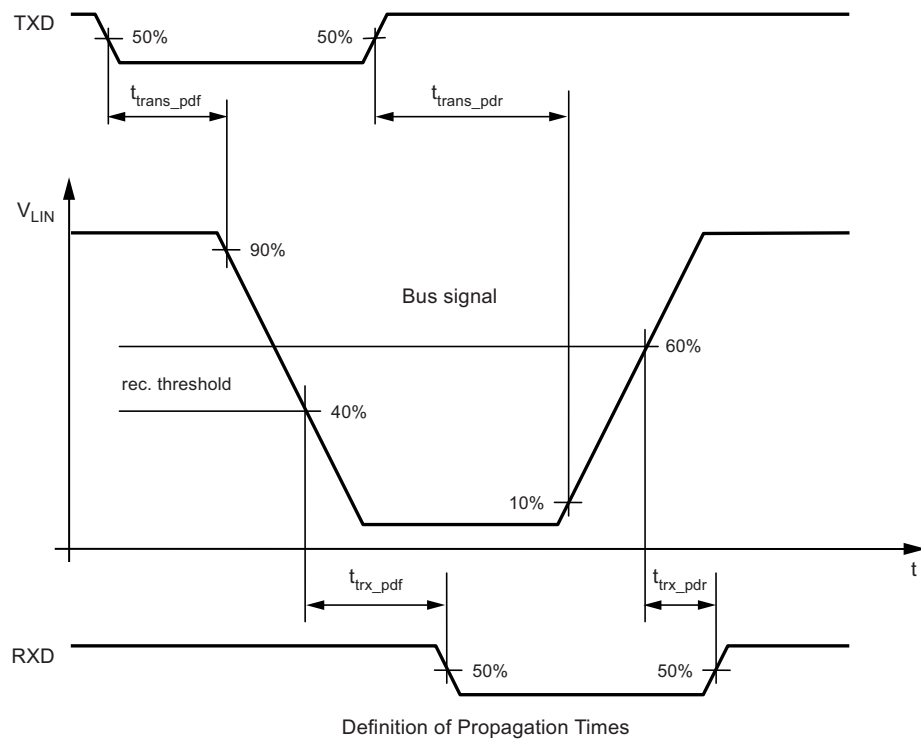
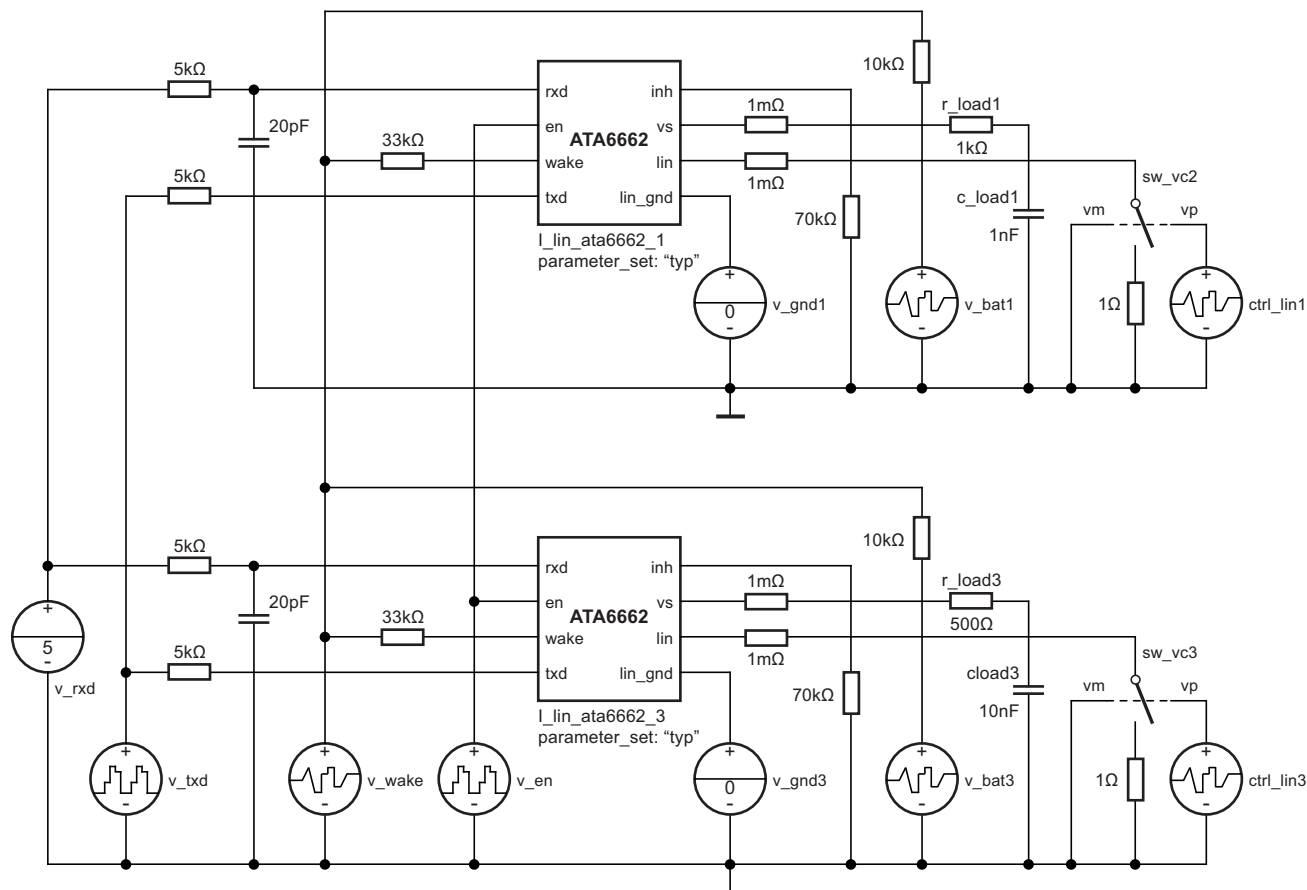


Figure 6-3. Test Bench lin_ata666x_tb_7_18 ($V_{bat1} = 7V$, $R_{load1} = 1k\Omega$, $C_{load1} = 1nF$, $V_{bat3} = 18V$, $R_{load3} = 500\Omega$, $C_{load3} = 10nF$)



6.2 Simulation Options

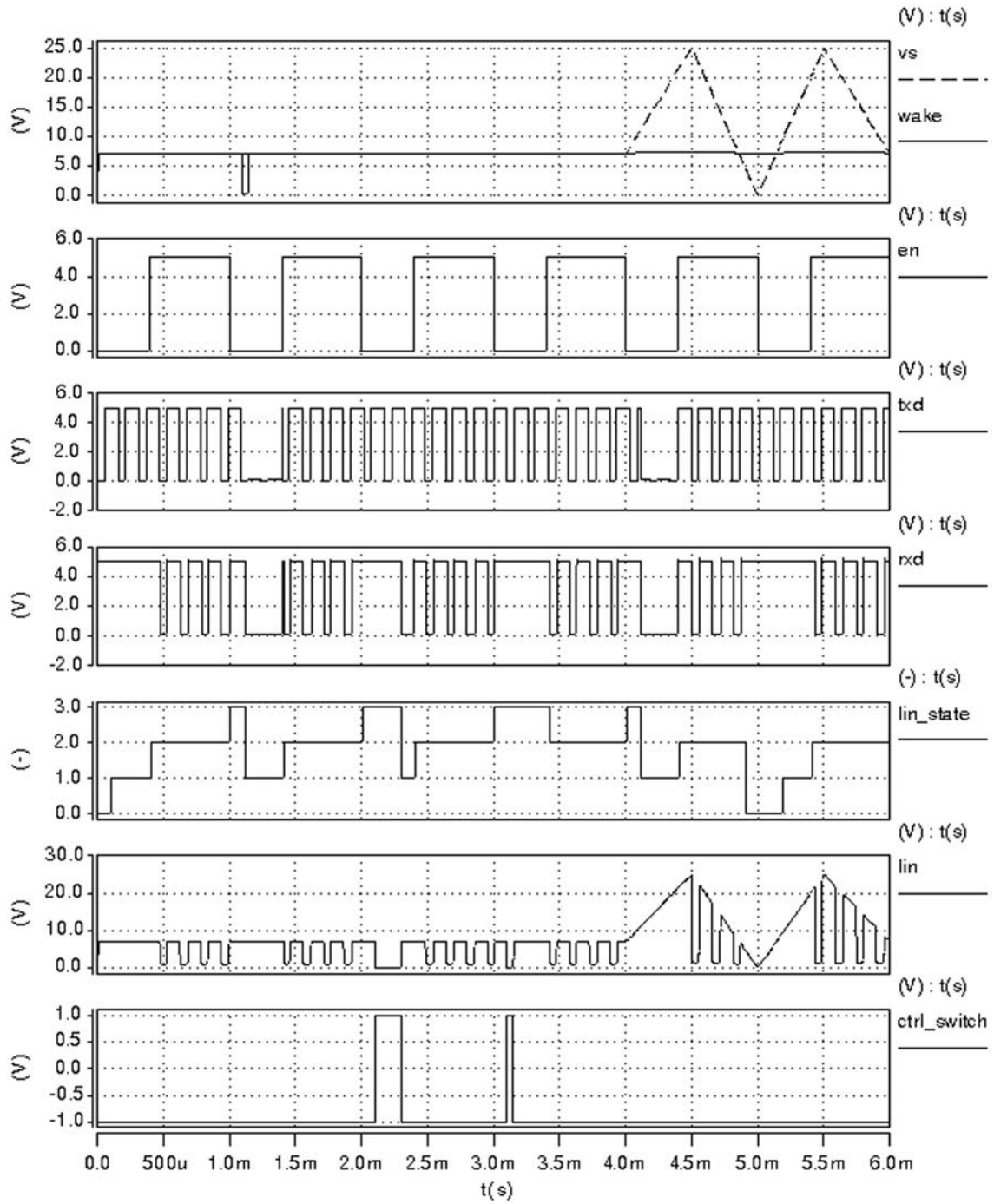
During transient simulations the following settings have to be used for each test bench to see all signal changes:

End Time [sec], default:	6m	
End Time [sec], lin_ata666x_tb_6662_6663:	45m	
Time Step [sec]:	1n	(AdvanceMS only)

The other settings can be left at their values.

6.3 Simulation Results

Figure 6-4. Signal Changes in Test Bench lin_ata666x_tb, Atmel ATA6662



The LIN transceiver is off ($\text{lin_state}=0$) if $V_S < v_{\min}$. This behavior predominates. If V_S rises higher than $v_{\min}$, the transceiver goes to pre-normal/fail-safe mode ($\text{lin_state} = 1$), and the other signals are interpreted.

Then the LIN transceiver goes to normal mode ($\text{lin_state} = 2$) if EN is high. The LIN transceiver goes to sleep mode ($\text{lin_state} = 3$) if EN is low (open).

The LIN transceiver can awake from sleep mode either when $\text{wake} + \text{add_wakel} < V_S$ or by a strong pull-down at pin LIN (the model uses timers for t_{wake} and t_{bus} respectively). The strong pull-down at pin LIN is realized by a switch when ctrl_switch equals 1.

Figure 6-5. Response to an Impulse at TXD in Test Bench lin_ata666x_tb_vary with Different Parameter Settings, Atmel ATA6662

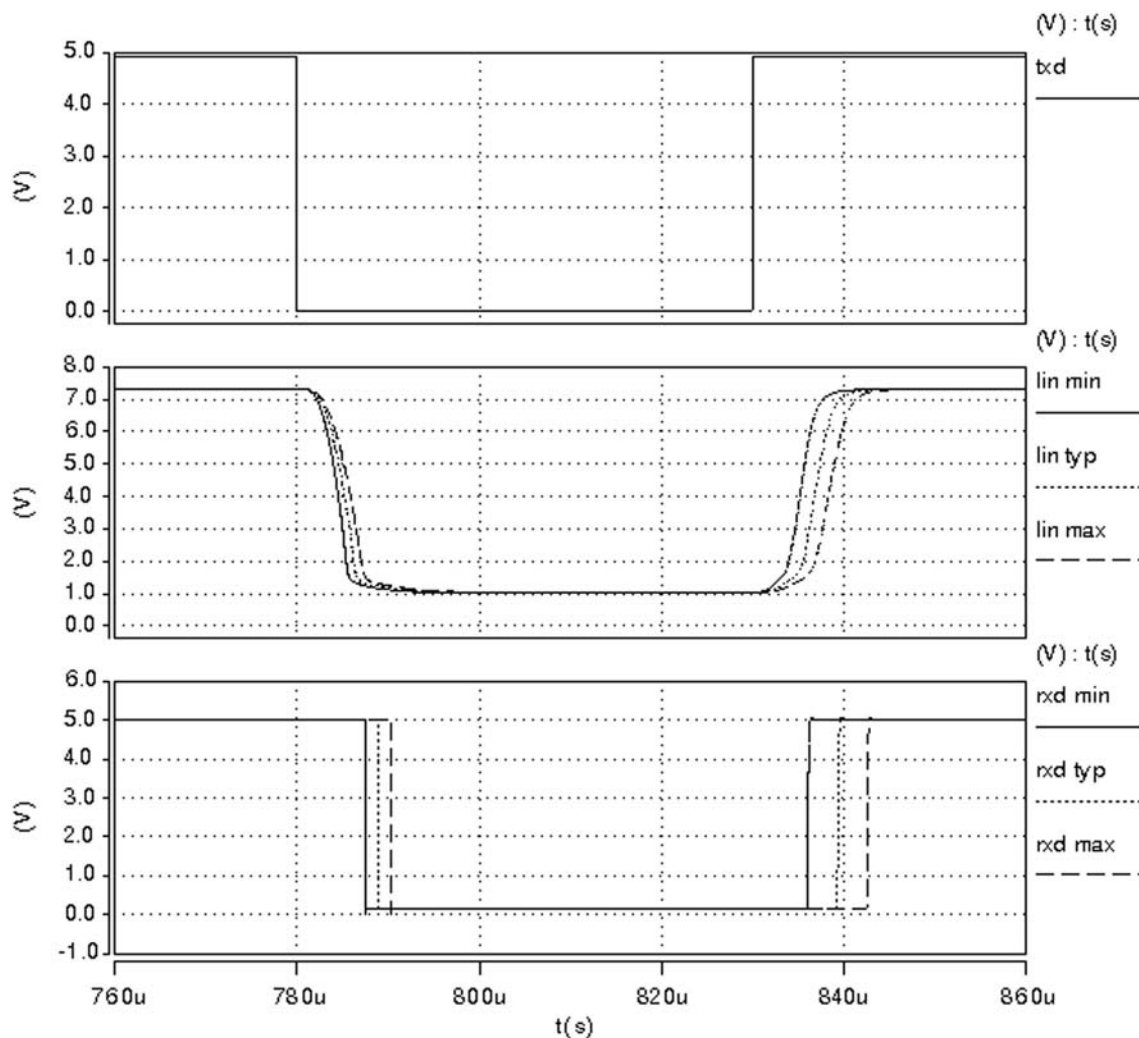
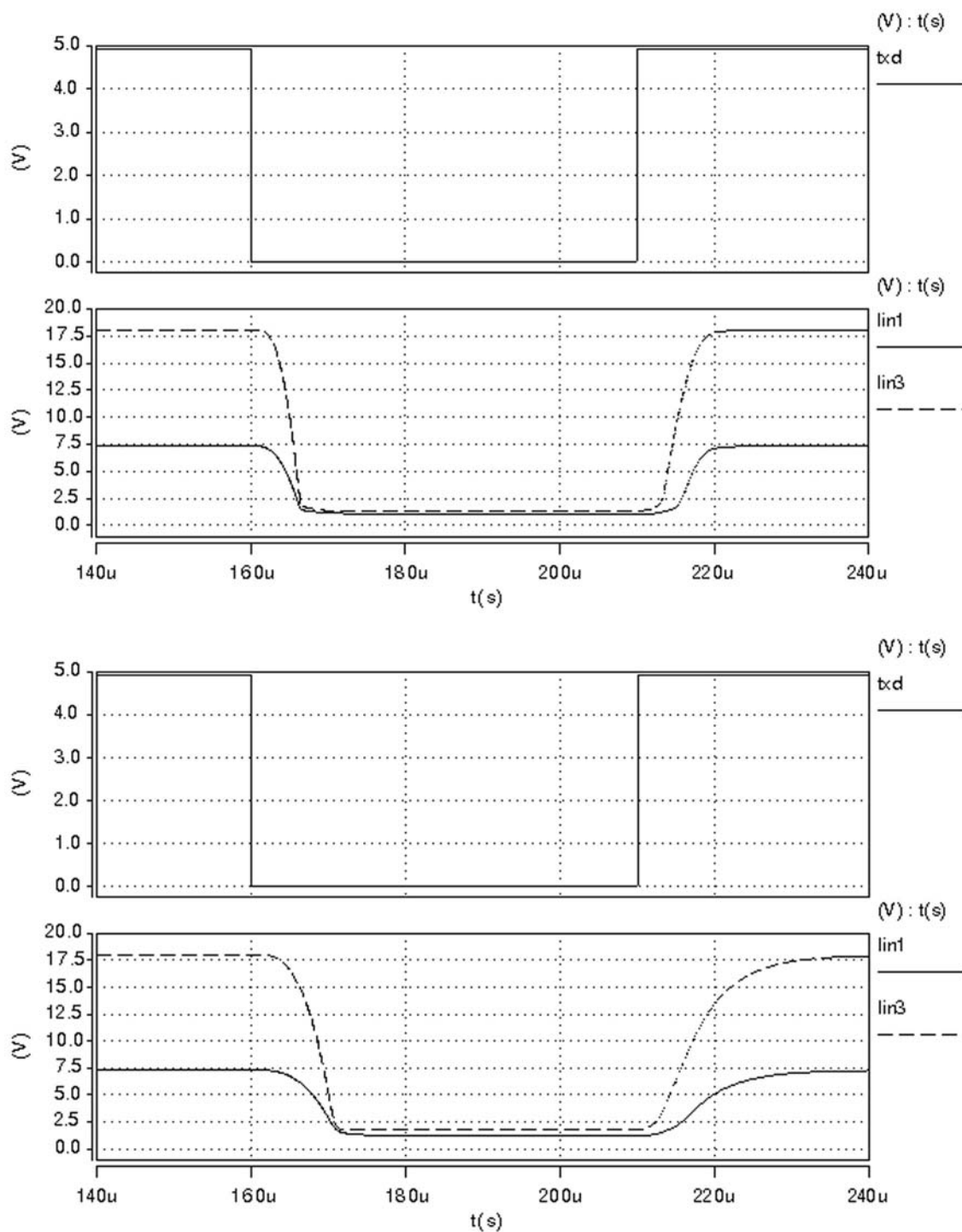


Figure 6-6. Response to an Impulse at TXD in Test Bench lin_ata666x_tb_7_18 with Different Load
(Top: $R_{load} = 1k\Omega$, $C_{load} = 1nF$; bottom: $R_{load} = 500\Omega$, $C_{load} = 10nF$), Atmel ATA6662



7. Availabilities

The following files are available:

Table 7-1. LIN Model

Filename	Description
lin_ata666x.vhd	LIN transceiver (macro model)
lin_parameters.vhd	Prepared parameter sets
lin_control.vhd	State machine of the LIN transceiver
lin_core.vhd	LIN core (macro model)
lin_core_control.vhd	Trigger for the LIN driver; in addition receiver and LIN wake detection
lin_slew_rate_control.vhd	Realizes the slew rate of the LIN impulse
lin_pullup.vhd	LIN pull-up resistor
lin_inh_driver.vhd	Inhibit switch
lin_lin_driver.vhd	LIN driver
lin_rxd_driver.vhd	RXD driver
lin_txd_driver.vhd	TXD pull-down transistor
lin_power_supply.vhd	Internal power supply
lin_bias.vhd	Bias circuit inside the LIN core
lin_current.vhd	Current source (becomes a resistor if the voltage is lower than $v_{min} = 0.1V$)
lin_current_control.vhd	Current source controlled by state and wake-up (current consumption of IC)
lin_wake_current.vhd	Wake current source
lin_watchdog.vhd	Watchdog (not part of the LIN transceiver's behavioral model)
and2.vhd	An and gate with 2 inputs
set_state.vhd	Forces a state to a given value
inv_state.vhd	Inverter

Table 7-2. Test Benches

Filename	Description
primitives_tb.vhd	Primitives for all test benches
lin_ata666x_tb.vhd	General functional test
lin_ata666x_tb_7_18.vhd	Compares two test circuits powered by 7V and 18V
lin_ata666x_tb_A.vhd	General functional test
lin_ata666x_tb_B.vhd	Tests local wake-up request
lin_ata666x_tb_C.vhd	Tests wake-up via LIN (remote wake-up request)
lin_ata666x_tb_D.vhd	Compare three test circuits, parameter_set min, typ, and max
lin_ata666x_tb_E.vhd	Test TXD time-out function of ATA6662, ATA6663, and ATA6664
lin_ata666x_tb_F.vhd	Tests wake-up via floating LIN (ATA6662 and ATA6663)
lin_ata666x_tb_G.vhd	Tests current consumption (resistors have high impedance)
lin_ata666x_tb_vary.vhd	Compares three test circuits, parameter_set min, typ, and max
lin_ata666x_tb_6662_6663.vhd	Compares ATA6662 and ATA6663

All test benches except the comparison tests *lin_ata666x_tb_E.vhd*, *lin_ata666x_tb_F.vhd*, and *lin_ata666x_tb_6662_6663.vhd* use the nominal setting of parameter *lin_type* preset in *lin_ata666x.vhd*. If *lin_type* equals "6663," the test benches are based on the ATA6663 model, if *lin_type* equals "6664," the test benches are modeled on the ATA6664. The ATA6662 serves as the model in all other cases.

8. Revision History

Please note that the following page numbers referred to in this section refer to the specific revision mentioned, not to this document.

Revision No.	History
9205D-AUTO-06/15	• Put document in the latest template
9205C-AUTO-11/11	• Section 4 “Using the Model” on page 10 updated
9205B-AUTO-03/11	• C versions on page 1 added

