

**LogicSimulator.doc**

**COLLABORATORS**

	<i>TITLE :</i> LogicSimulator.doc		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		June 8, 2025	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>LogicSimulator.doc</b>	<b>1</b>
1.1	main . . . . .	1
1.2	introduction . . . . .	1
1.3	distribution . . . . .	2
1.4	requirements . . . . .	3
1.5	installation . . . . .	3
1.6	gates . . . . .	4
1.7	menus . . . . .	4
1.8	gadgets . . . . .	5
1.9	menu_gates . . . . .	6
1.10	new . . . . .	6
1.11	load . . . . .	7
1.12	save . . . . .	7
1.13	saveas . . . . .	7
1.14	saveiff . . . . .	7
1.15	print . . . . .	7
1.16	comment . . . . .	8
1.17	about . . . . .	8
1.18	redraw . . . . .	8
1.19	quit . . . . .	8
1.20	movecirc . . . . .	8
1.21	connect . . . . .	8
1.22	addknot . . . . .	9
1.23	removeknot . . . . .	9
1.24	removegate . . . . .	9
1.25	removewire . . . . .	9
1.26	undo . . . . .	10
1.27	label . . . . .	10
1.28	copy . . . . .	10
1.29	cut . . . . .	11

---

---

1.30	start	11
1.31	stop	11
1.32	pause	11
1.33	highlight	11
1.34	insert	12
1.35	screenmode	12
1.36	workarea	12
1.37	usewb	12
1.38	ewinopen	12
1.39	savewithicon	12
1.40	saveset	13
1.41	help	13
1.42	contents	13
1.43	gate_xinputs	13
1.44	gate_switch	13
1.45	gate_led	14
1.46	gate_and	14
1.47	gate_nand	15
1.48	gate_or	16
1.49	gate_nor	17
1.50	gate_xor	18
1.51	gate_not	19
1.52	gate_rsff	20
1.53	gate_jkff	21
1.54	gate_jkmsff	22
1.55	gate_jkmsrff	23
1.56	gate_tff	25
1.57	gate_trsff	26
1.58	gate_dff	27
1.59	gate_clock	28
1.60	gate_mono	28
1.61	gate_tond	29
1.62	gate_toffd	30
1.63	gate_trafficlight	31
1.64	gate_numdisplay	31
1.65	gate_inputinv	32
1.66	gate_low	32
1.67	gate_high	32
1.68	timerequester	32

---

---

1.69	switchrequester . . . . .	33
1.70	ledrequester . . . . .	33
1.71	trafficlightrequester . . . . .	33
1.72	numdisplayrequester . . . . .	34
1.73	labelrequester . . . . .	34
1.74	author . . . . .	34
1.75	credits . . . . .	35
1.76	keyboard . . . . .	35
1.77	hints . . . . .	36

---

## Chapter 1

# LogicSimulator.doc

### 1.1 main

Logic Simulator

A program to simulate logic circuits

Version 1.0

Copyright (c) 1995/96 by Andreas Tetzl

Introduction	What is it ?
Distribution	Legal stuff
System requirements	What do you need
Installation	How to install it on Harddisk
Gates	The supported Gates
Menus	The Menus
Gadgets	The Buttons
Hints	Please read this !
The Author	How to contact the author
Keyboard equivalents	Shortcuts
Thanks	Acknowledgements

### 1.2 introduction

LogicSim is a program to simulate digital logic circuits. It has a full graphic user interface to design and edit the circuits.

---

#### Features:

- unlimited number of gates
- user definable work space
- AND, OR, NAND, NOR and XOR gates with 2, 3 or 5 inputs
- RS, JK, JKMS, T and D flip flops
- JKMS and T FlipFlops with independent Set and Reset inputs
- user definable Timer: Clock, Monoflop, turn on delay and  
turn off delay
- saving the circuit as iff or print it
- input inverter
- traffic light model
- 7 segment display in HEX or decimal
- highlighting of wires with HIGH while simulating
- replacing "compatible" gates without removing them first
- you can label your gates with selectable position and font
- all windows are font sensitive and resizable
- UNDO function
- AmigaGuide online help with diagrams
- it's localized (currently english and german)
- copy/cut/insert function

### 1.3 distribution

LogicSimulator is written and copyright (c) 1995/1996 by  
Andreas Tetzl.

This program is mail-ware !  
Please write me your opinion about the  
program, suggestions or bug reports.

This is NOT public domain !

LogicSim may be freely distributed for non-commercial purposes,  
as long as no files of the archive are changed or removed.

No charge may be made for LogicSim, except cost for media,  
copying or downloading.

Commercial use or inclusion in other software packages  
is explicitly prohibited without the permission of the author !

Contact me, if you want to use it in any  
commercial way !

No warranties are made for this program. Use it at your own risk !

---

## 1.4 requirements

Minimum requirements:

- An Amiga (:-)
- Amiga OS 2.04 or higher
- 1 MB chip RAM

I recommend a faster CPU, additional fast memory and OS 3.0 for serious use.

If present, LogicSim uses many features of OS 3.0 (pen-sharing, locale support).

LogicSim uses the ASL.library for File-, Font- and ScreenMode-Requesters. If reqtools.library is installed, LogicSim uses the FileRequester of this library. If your asl.library doesn't have a ScreenMode-requester (v37), the ScreenMode requester of reqtools-library is also used.

## 1.5 installation

The program is useable without any installation.

But the best way is to use the supplied installer script.

It will copy the main program, the docs, the font, the catalog and example circuits to your harddisk and it creates the show script for the documentation.

No changes are made to s:user-startup.

The configuration is saved in the same directory as the main program, not in ENVARC or S:.

LogicSim can be started from Shell or Workbench.

From Workbench it is also started when you double

---

click a icon of a circuit in the Circuits directory.  
It will open the selected circuit.

## 1.6 gates

basic gates

AND  
NAND  
OR  
XOR  
NOR  
NOT

Switch  
LED

Flip Flops

RS  
JK  
JKMS  
JKMS (S/R)  
T  
T (S/R)  
D

Timer

Clock  
MonoFlop  
turn on delay  
turn off delay

Special

traffic light  
numeric display  
input inverter  
HIGH  
LOW

## 1.7 menus

Project	Edit	Simulate	Settings	Help
New	basic gates »	Start	ScreenMode...	Help

Load...	Flip-Flops	»	Stop	Work area...	Contents
Save	Timer	»	Pause	Use Workbench	
Save as...	Special	»	highlight HIGH	edit window open	
Save IFF...	Input	»		save with icon	
Print...	-----			Save settings	
Comment...	Connect				
About...	Remove gate				
Redraw	Remove wire				
Quit	Add knot				
	Remove knot				
	Move Circuit				
	Label				
	Undo				
	-----				
	Copy				
	Cut				
	Insert				

## 1.8 gadgets

"Edit-Window"

The gadgets in this window are used to open and close the other windows below.

If you have all other windows opened, you can close this window. You can reopen it in the settings menu.

"Actions-Window"

Connect  
Remove gate  
Remove wire  
Start  
Stop  
Undo

"basic gates"

AND  
NAND  
OR  
NOR  
XOR  
NOT  
Switch  
LED  
2  
3  
5

"FlipFlops"

```

RS-FlipFlop
JK-FlipFlop
JKMS-FlipFlop
JKMS-FlipFlop (S/R)
D-FlipFlop
T-FlipFlop
T-FlipFlop (S/R)

```

"Timer"

```

Clock
Monoflop
turn on delay
turn off delay

```

"Special"

```

traffic light
numeric display
input inverter
HIGH
LOW

```

## 1.9 menu\_gates

Select one of the menu items to place a gate in your circuit.

basic gates	FlipFlops	Timer
AND	RS	Monoflop
NAND	JK	Clock
OR	JKMS	turn on delay
NOR	JKMS (S/R)	turn off delay
XOR	D	
NOT	T	
Switch	T (S/R)	
LED		

```

-----
2 inputs
3 inputs
5 inputs

```

Special	Input
traffic light	input inverter
numeric display	HIGH
	LOW

## 1.10 new

---

Delete the old circuit and begin a new one.

You will be asked to save the circuit first.

## **1.11 load**

Load a circuit from disk.  
The old circuit will be deleted.

You will be asked to save the old circuit first.

If the size of the loaded circuit is greater than your current work area size, the work area size will be increased.

## **1.12 save**

Save a circuit to disk.

If no filename was given before, a filerequester will be opened.

## **1.13 saveas**

This equals the Save menu, but you will be asked for a filename first.

## **1.14 saveiff**

This allows you to save your circuit as IFF-ILBM picture.

The circuit is saved as uncompressed ILBM. I hope all programs are able to load this.

## **1.15 print**

This allows you to print your circuit.

You will be asked to open the GfxPrefs-Editor.  
You can also print an additional Comment.

---

## 1.16 comment

You can enter an additional comment, author and date to your circuit. This comment can be printed with the circuit.

## 1.17 about

This shows some information about the program.

If you have OS3.0+ and at least 32 colors, a picture is shown in the about requester. The picture is stored in LSAboutPic. If you have no OS3, not enough colors or LogicSim can't find the picture file, an internal picture is shown.

## 1.18 redraw

This refreshes the circuit graphics.  
Not really needed, but I used it while developing.

## 1.19 quit

This item leaves the program.

You will be asked to save the circuit first.

## 1.20 movecirc

You can move the circuit in the work area.

A frame is drawn around the circuit while moving it. You can move this frame to the new position and press the left mouse button. The circuit will be moved to the new position.

## 1.21 connect

Shortcut: c

Connect a output of a gate or a knot with the input of a gate.

---

First you have to click on a output of gate or on a knot. Then you can draw a wire to an input of a gate. By pressing SPACE you can add a knot to the last position of the wire, by pressing 'u' you can remove the last drawn part of the wire.

You can exit the connect mode by pressing ESC or selecting an other menu item.

## 1.22 addknot

You can add a knot to a previous drawn wire.

Click on the wire where you want to add the knot. If you click near to an edge, the knot will be added to this edge.

## 1.23 removeknot

You can remove a knot from a wire by clicking on the knot to delete.

## 1.24 removegate

Shortcut: d

You can remove a gate from your circuit by calling this function and clicking on the gate to remove.

All wires connected to the removed gate will also be removed.

You can undo this action by pressing 'u' or the 'undo' button.

## 1.25 removewire

Shortcut: w

You can remove a wire from your circuit by calling this function and clicking on the wire to remove.

---

You can undo this action by pressing 'u' or the 'undo' button.

## 1.26 undo

Shortcut: u

With the UNDO function you can undo the last action (remove gate, remove wire...).

Using UNDO again will redo the action.

In future versions of LogicSim I'll make more undo-steps.

## 1.27 label

This menu item starts the label mode.

In this mode you can click on a gate to open the label-requester.

In this requester you can input a text, select its position and select a font for the text.

After this the gate will be labeled with your text.

You can exit the label mode by pressing the ESC-key.

## 1.28 copy

You can copy a part of your circuit to insert it later at another position or in a new circuit.

This function is similar to to the copy function in text editors.

All wires which are with beginning and end in the area you selected will be copied, too.

---

## 1.29 cut

This function is similar to the copy function, but all copied gates and wires are removed from your circuit afterwards.

## 1.30 start

This will start the simulation of your circuit.

You can pause the simulation with Pause.  
The simulation can be stopped with ESC or Stop.

Select the highlight HIGH wire menu, if you want to 'debug' your circuit.

## 1.31 stop

This will exit the simulation mode. It is the same as pressing the ESC key while simulating.

## 1.32 pause

Whit this menu item, you can halt your simulation in the current state. To run it again, turn this menu item off.

You can also use this menu item before starting the simulation with Start. NOTHING will be simulated, when you start the simulation in that case, until you turn the pause-item off.

## 1.33 highlight

When using this function, all wires with HIGH will be drawn white in simulation mode.

This is useful if you want to 'debug' your circuit.

This menu-item must be turned on/off BEFORE starting the simulation mode.

---

### 1.34 insert

With this function you can insert a previously copied or cutted part of a circuit in your actual circuit.

### 1.35 screenmode

You can select a ScreenMode in the ASL/ReqTools ScreenMode-Requester, LogicSim will run on.

Pressing the OK button in the requester will open a screen in the selected mode. You can move LogicSim back on the Workbench Screen by turning the menu use workbench on.

### 1.36 workarea

You can select the width and height of the work area (the area where you draw your circuits, it can be scrolled around).

Note: Big sizes will need \*MUCH\* chip-memory.

You can calculate the chip-mem needed by  $mem = width * height / 4$ .

If you set values which are bigger than your free chip-mem, the size will be decreased.

### 1.37 usewb

The display mode of LogicSim is changed from custom screen to Workbench screen and back.

### 1.38 ewinopen

If you have closed the edit-window (-> Windows), you can reopen it with this menu.

### 1.39 savewithicon

This lets LogicSim save your circuits with additional icons.

When you click these icons on workbench, LogicSim will be started and loads the circuit.

---

## 1.40 saveset

This will save your settings and window positions in the file LogicSim.config.

## 1.41 help

This tells you that the program has an online help system, if you don't know it. :-)

## 1.42 contents

This loads the contents page of this AmigaGuide document.

## 1.43 gate\_xinputs

Shortcut: 2, 3, 5

Here you can set the number of inputs for AND, NAND, OR, NOR, and XOR gates.

Possible numbers of inputs are 2, 3 and 5.

## 1.44 gate\_switch

Switch

Shortcut: s

This is an element for user inputs.

It can be used as switch or key. If you use it as switch, you can click on it (in simulation mode) to turn it on (output=HIGH) and click again to turn it off (output=LOW). If you use it as key, clicking on it turns it on and releasing the button after this turns it off.

Show Gate

Click on the gate to open the switch requester.

---

## 1.45 gate\_led

LED

Shortcut: l

This is an output element, it is colored if HIGH and transparent if LOW.

Show Gate

## 1.46 gate\_and

AND

Shortcut: a

The output is only HIGH when ALL inputs are HIGH.

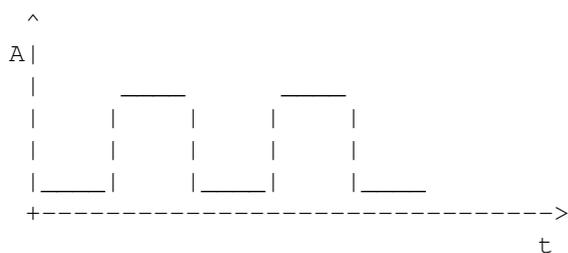
You can choose the number of inputs (2, 3 or 5) in the menu or the "basic gates"-window.

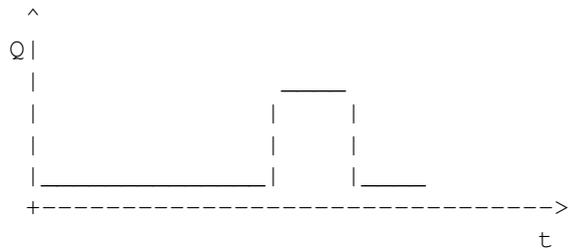
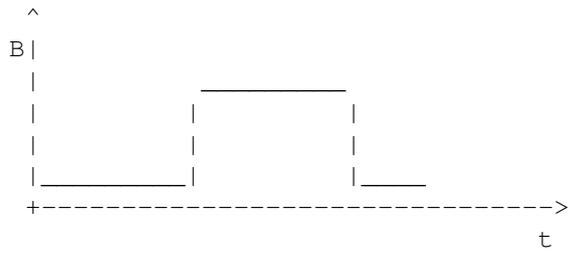
Click on the gate to replace it with NAND, OR, NOR, XOR.

A	B	Q	
0	0	0	A - input 1
0	1	0	B - input 2
1	0	0	Q - output
1	1	1	

Show Gate

Show diagram





### 1.47 gate\_nand

NAND

Shortcut: A

This is an AND with inverted output

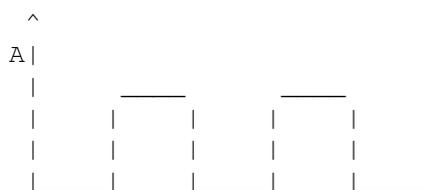
You can choose the number of inputs (2, 3 or 5) in the menu or the "basic gates"-window.

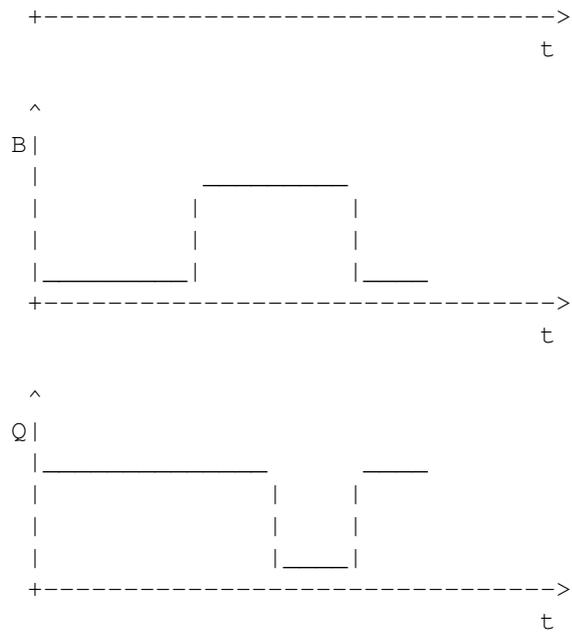
Click on the gate to replace it with AND, OR, NOR, XOR.

A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

A - input 1  
B - input 2  
Q - output

Show Gate  
Show diagram





## 1.48 gate\_or

OR

Shortcut: o

The output is HIGH when at least one of the inputs is HIGH.

You can choose the number of inputs (2, 3 or 5) in the menu or the "basic gates"-window.

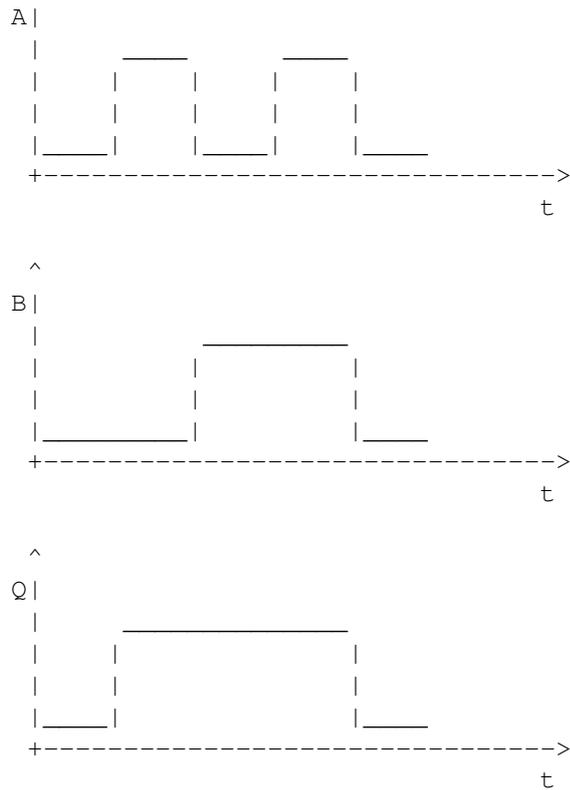
Click on the gate to replace it with AND, NAND, NOR, XOR.

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

A - input 1  
B - input 2  
Q - output

Show Gate  
Show diagram

^



## 1.49 gate\_nor

NOR

Shortcut: 0

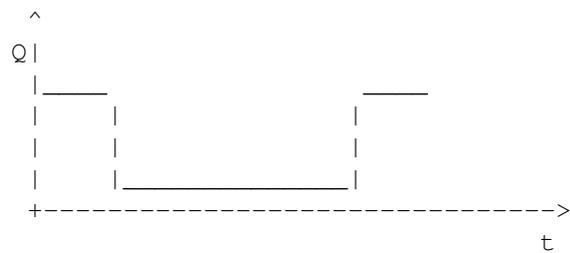
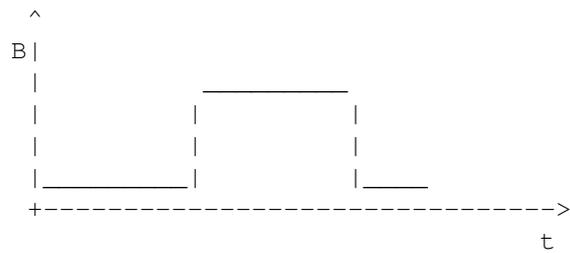
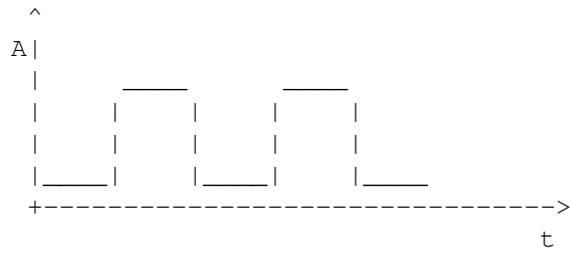
This is an OR with inverted output.

You can choose the number of inputs (2, 3 or 5) in the menu or the "basic gates"-window.

Click on the gate to replace it with AND, NAND, OR, XOR.

A	B	Q	
0	0	1	A - input 1
0	1	0	B - input 2
1	0	0	Q - output
1	1	0	

Show Gate  
 Show diagram



## 1.50 gate\_xor

XOR

eXclusive OR

Shortcut: x

The output is HIGH when only ONE of the inputs is HIGH.

You can choose the number of inputs (2, 3 or 5) in the menu or the "basic gates"-window.

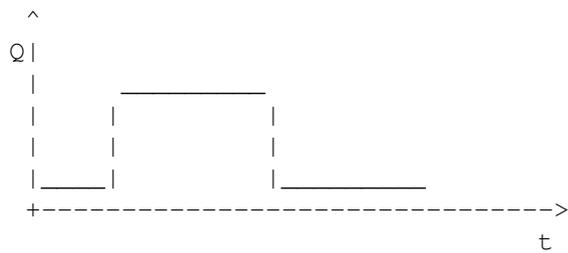
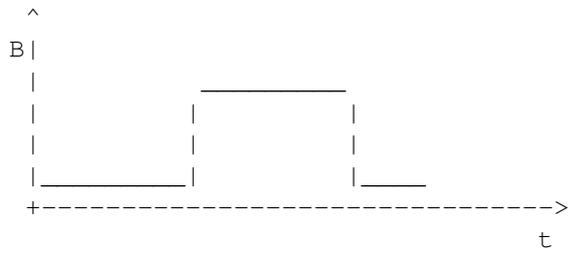
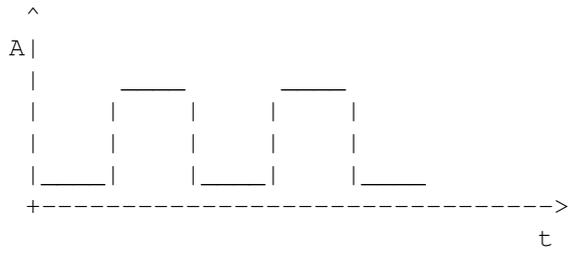
Click on the gate to replace it with AND, NAND, OR, NOR.

A	B	Q
0	0	0

A - input 1

0		1		1	B - input 2
1		0		1	Q - output
1		1		0	

Show Gate  
 Show diagram



## 1.51 gate\_not

NOT

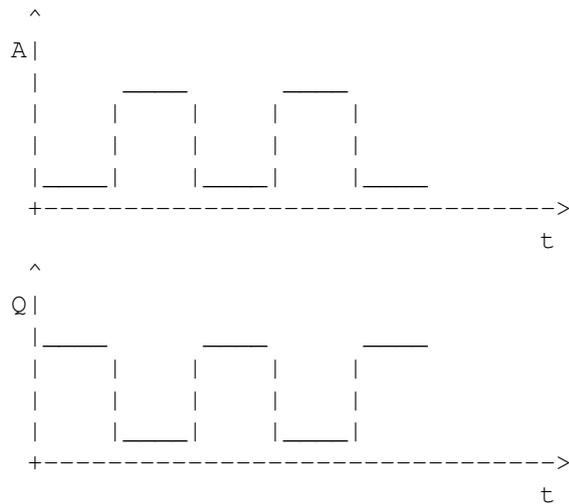
Shortcut: n

The output is an inverted input.  
 If input is LOW, output is HIGH.  
 If input is HIGH, output is LOW.

A | Q  
 ---+---

0		1		A - input
1		0		Q - output

Show Gate  
Show diagram



## 1.52 gate\_rsff

RS FlipFlop

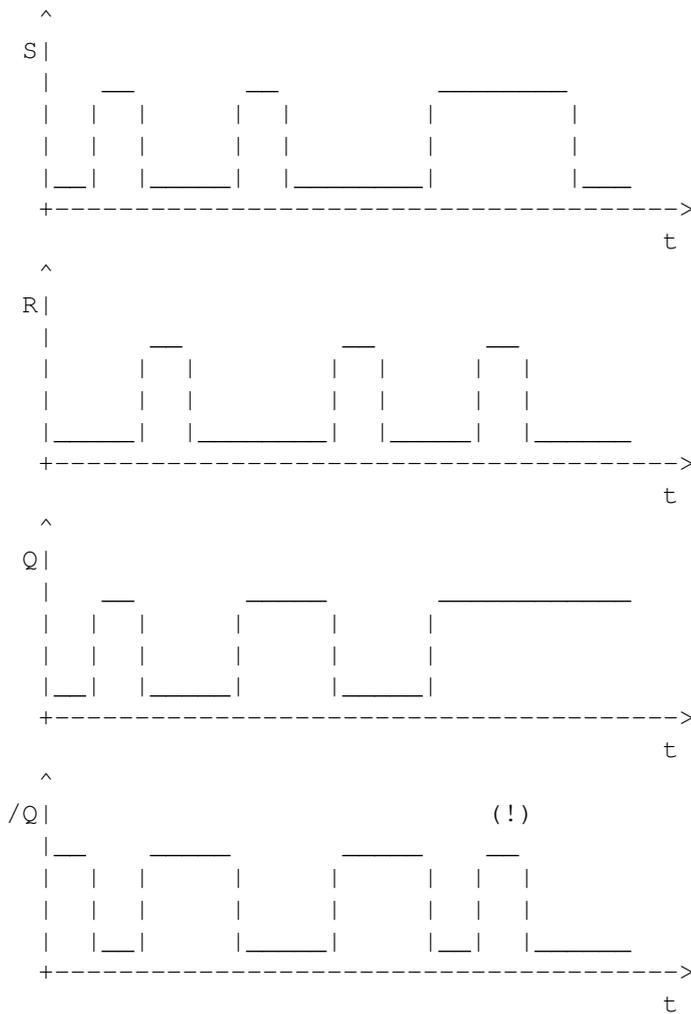
S = Set  
R = Reset  
Q = Output  
/Q= inverted Output

The output Q is set to HIGH by S=HIGH and reset to LOW by R=HIGH.  
The output state is kept even if S or R go back to LOW.

/Q is an inverted output Q.

S=R=HIGH is NOT allowed, because it produces an illegal output state Q=/Q=HIGH. Have a look at JK-FlipFlop.

Show Gate  
Show diagram



### 1.53 gate\_jkff

JK FlipFlop

J = Set

K = Reset

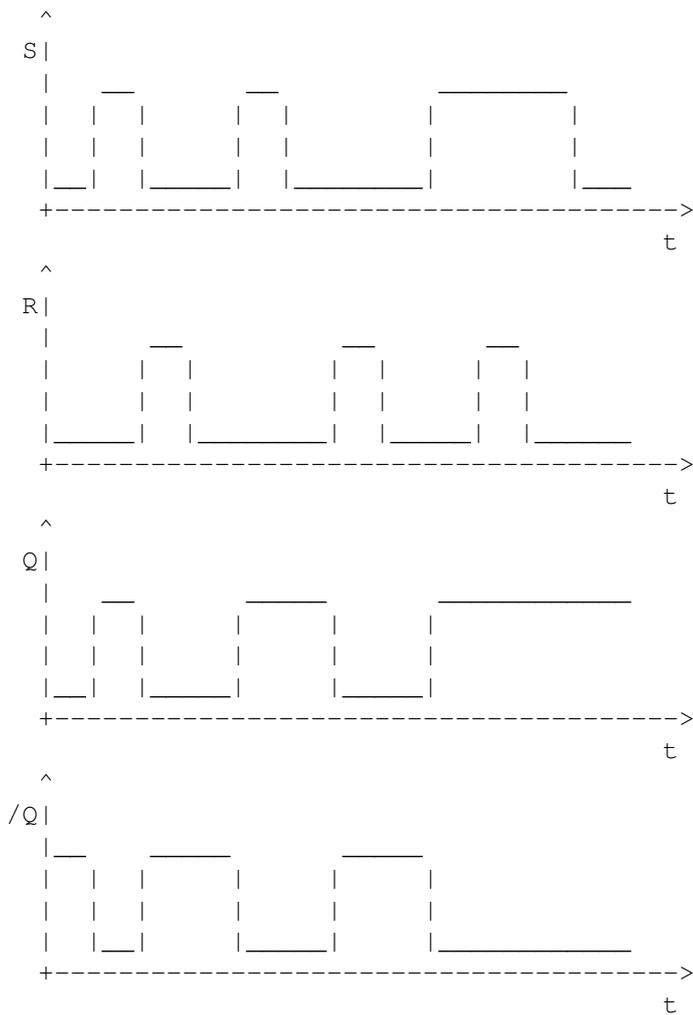
Q = Output

/Q = inverted Output

This FlipFlop is similar to the RS-FlipFlop but the problem with R=S=HIGH is solved.

Show Gate

Show diagram



## 1.54 gate\_jkmsff

JKMS FlipFlop

JK-Master-Slave FlipFlop

J = Set

K = Reset

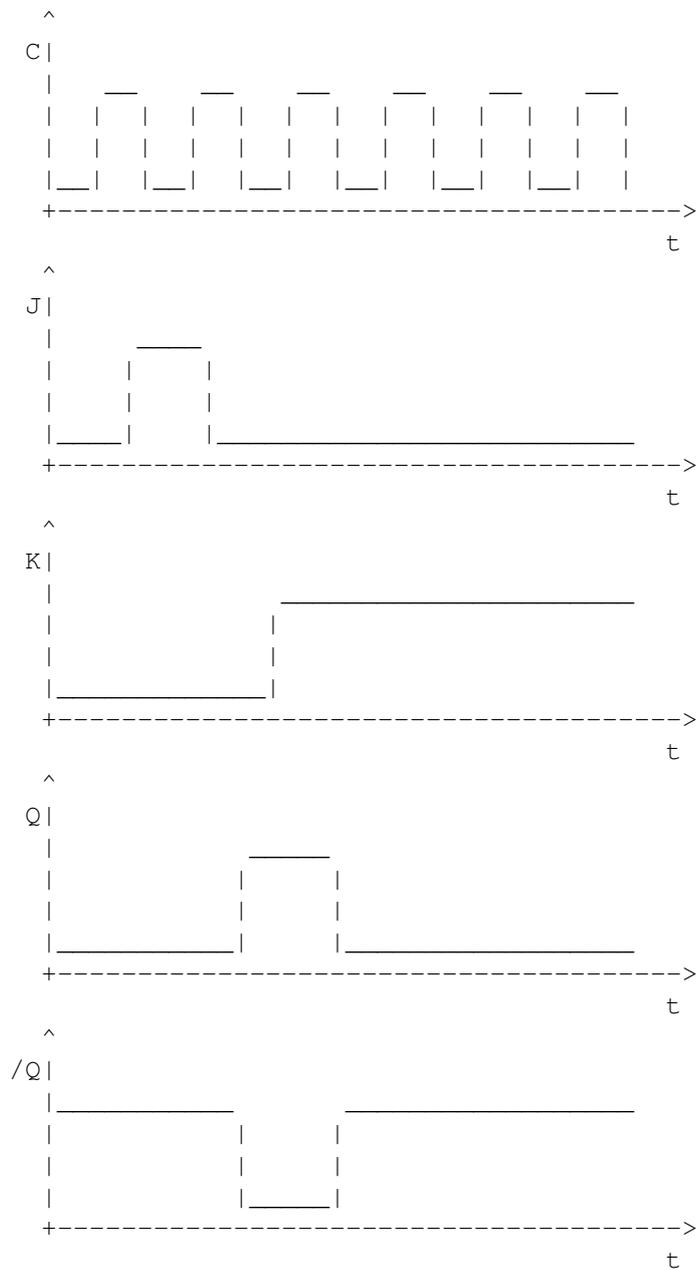
C = Clock

Q = Output

/Q= inverted Output

This type of FlipFlop consists internally of two JK-FlipFlops. The output is only changed when clock jumps from HIGH to LOW.

Show Gate  
 Show diagram



### 1.55 gate\_jkmsrsff

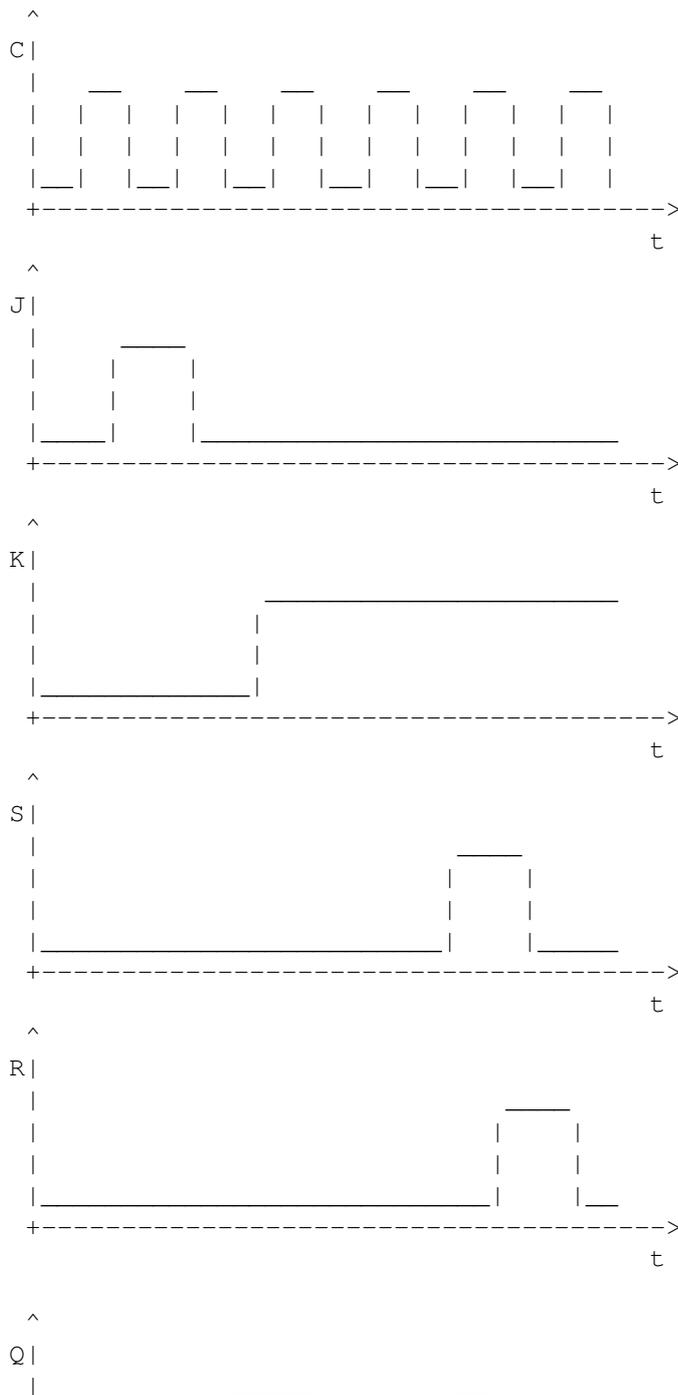
JKMS FlipFlop (S/R)

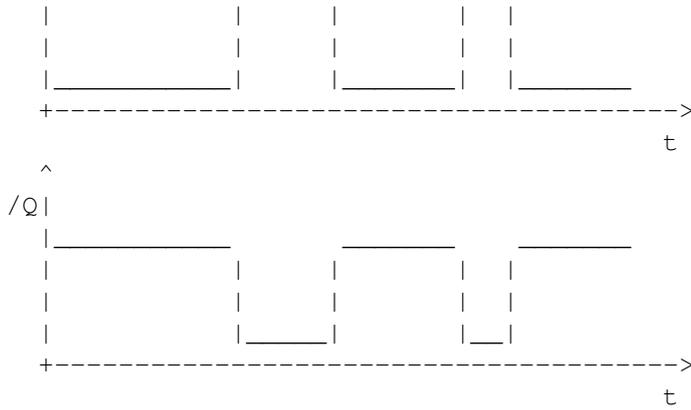
JK-Master-Slave FlipFlop with Set/Reset

This is the same as JKMS-FlipFlop, but it has special inputs to set and reset the output independent from the J, K and CLK inputs.

This can be used for example with counters that reset after the 10th impulse.

Show Gate  
Show diagram



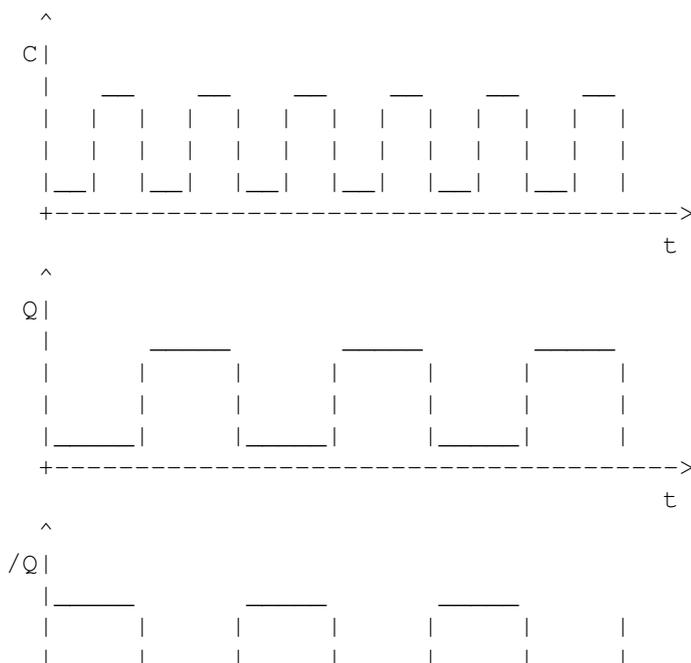


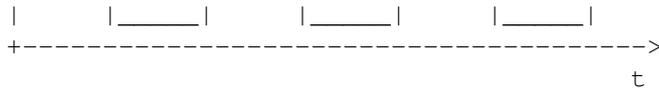
### 1.56 gate\_tff

T FlipFlop  
 C = Clock  
 Q = Output  
 /Q= inverted Output

This FlipFlop equals a JKMS-FlipFlop with J=K=HIGH.  
 The output is changed on each HIGH-LOW-jump of the clock impulse.

Show Gate  
 Show diagram





### 1.57 gate\_trsf

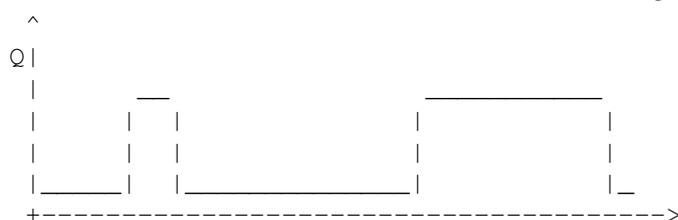
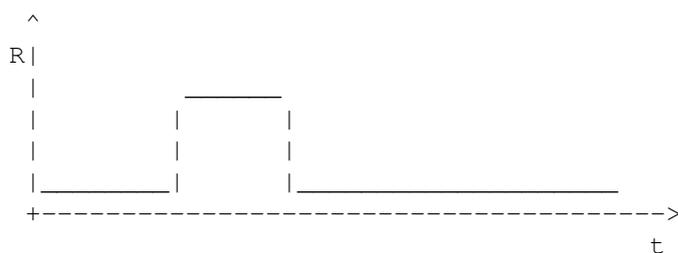
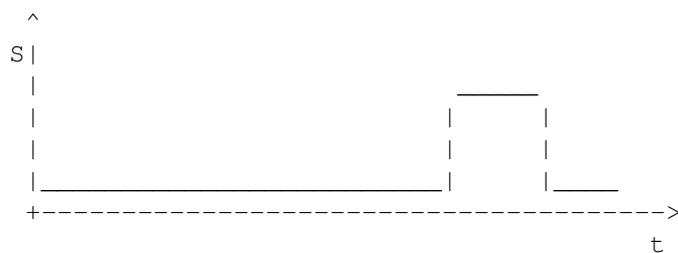
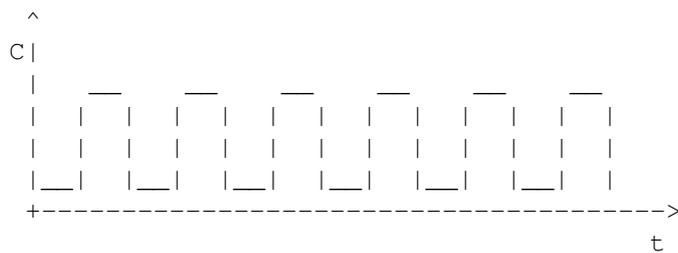
T FlipFlop (S/R)

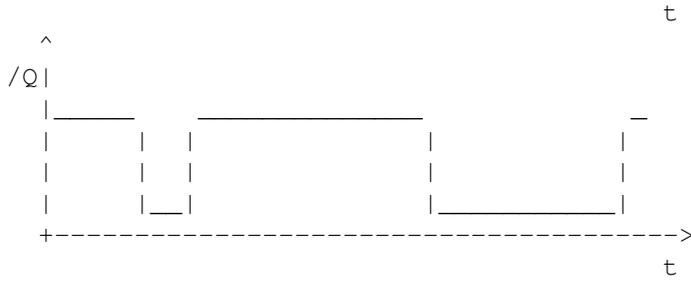
T FlipFlop with Set/Reset

This is the same as T-FlipFlop, but it has special inputs to set and reset the output independent from the J, K and CLK inputs.

This can be used for example with counters that reset after the 10th impulse.

Show Gate  
Show diagram





### 1.58 gate\_dff

Delay FlipFlop

C = Clock

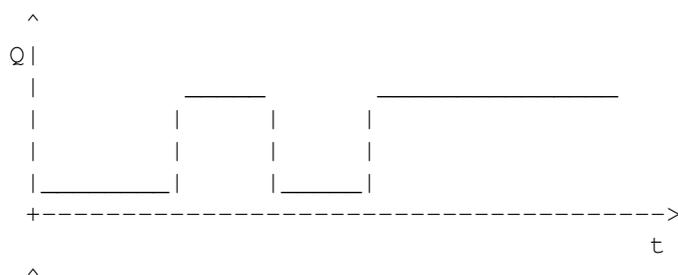
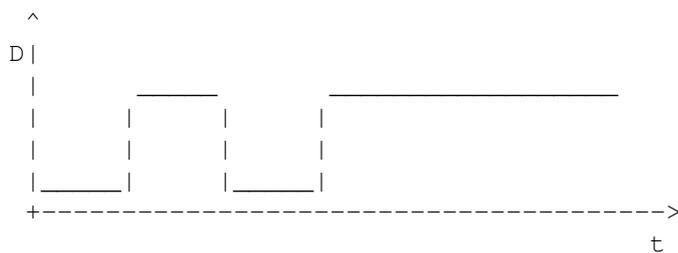
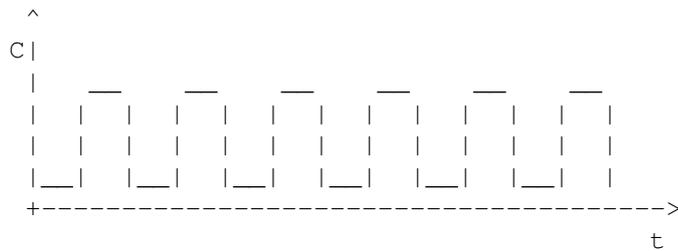
D = Input

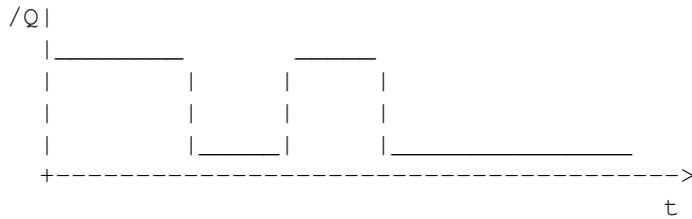
Q = Output

$/Q$  = inverted Output

The input is taken delayed to the output at the next positive clock.

Show Gate  
Show diagram



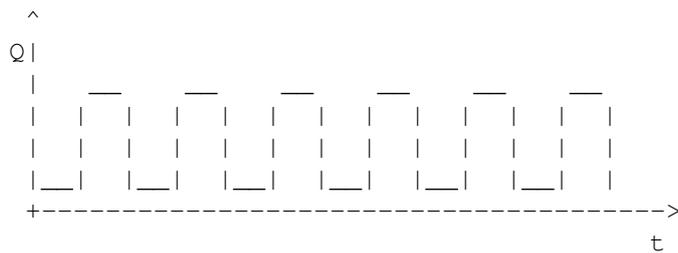


## 1.59 gate\_clock

Clock

This is a user adjustable clock generator. Click on the gate to open the time-requester where you can adjust impulse-time, pause-time the start level.

Show Gate  
Show diagram



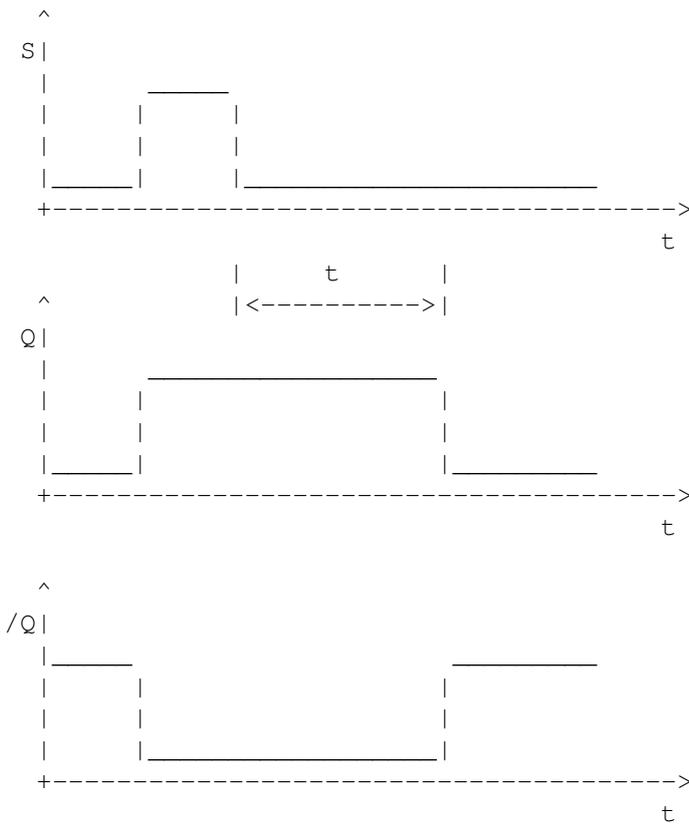
## 1.60 gate\_mono

Monoflop

This is a user adjustable Monoflop.  
The output goes to HIGH at the LOW-HIGH jump of the input.  
It falls back to low after the specified time is elapsed.

Click on the gate to open the time-requester where you can adjust the time.

Show Gate  
Show diagram



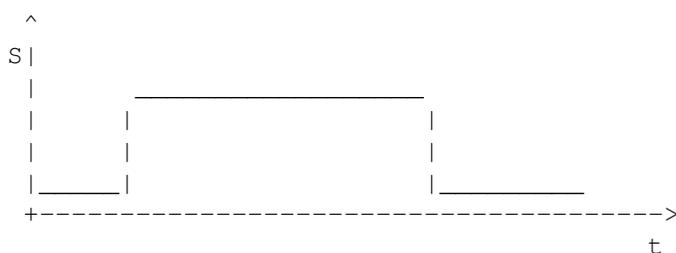
### 1.61 gate\_tond

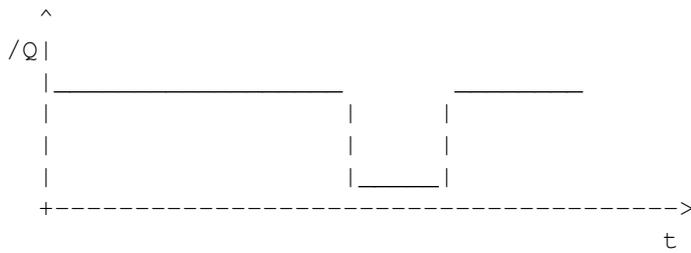
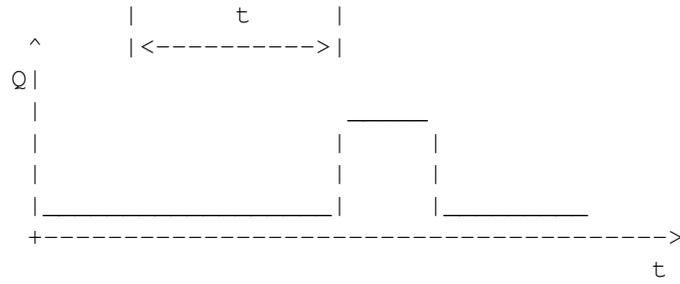
turn on delay

The output goes to HIGH after a specified time.

Click on the gate to open the time-requester where you can adjust the time.

- Show Gate
- Show diagram





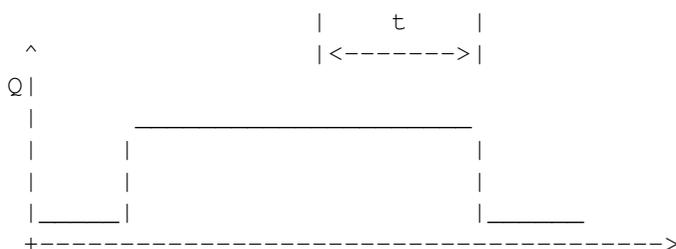
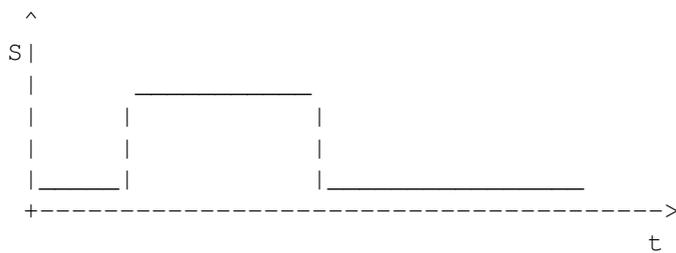
### 1.62 gate\_toffd

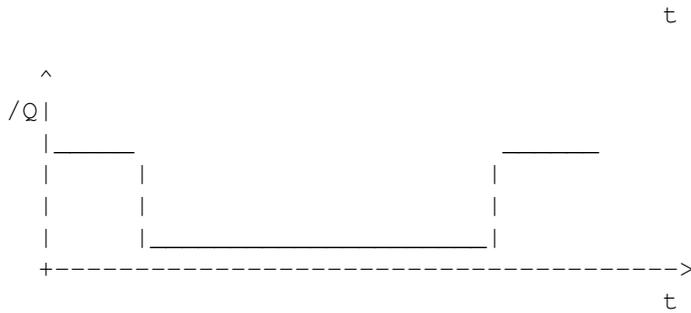
turn off delay

The output goes back to LOW after a specified time after the HIGH-LOW jump of the input.

Click on the gate to open the time-requester where you can adjust the time.

Show Gate  
Show diagram





### 1.63 gate\_trafficlight

traffic light model

This is a model of a traffic light with red, yellow and green lights.

Show Gate

Per default the color of each light is blue. Click on the gate to open the traffic light requester where you can select the color for each light.

### 1.64 gate\_numdisplay

7 segment display model

This is an model of a 7 segment numeric display. It shows decimal numbers from 0 to 31 or HEX numbers from \$0 to \$1F. It has 5 inputs that represent the binaries  $2^0 \dots 2^4$ . If you only want number up to 15 or 8, you should set the inputs  $2^5, 2^4 \dots$  to LOW.

Show Gate

Click on the gate to open a requester where you can choose between decimal and hexadecimal format.

## 1.65 gate\_inputinv

input inverter

This is the same as the NOT gate, but with graphically differences.

This gate is placed directly to an input of a gate. So the level on input is first inverted before it goes to the input of the 'big' gate.

Show Gate

## 1.66 gate\_low

LOW

This is to permanently set an input of a gate to LOW. For example, if you need a OR with 4 inputs, take one with 5 inputs and set one input to LOW.

Show Gate

## 1.67 gate\_high

HIGH

This is to permanently set an input of a gate to HIGH. For example, if you need a AND with 4 inputs, take one with 5 inputs and set one input to HIGH.

Show Gate

## 1.68 timerequester

This is the requester to set the time values of Clock, Monoflop, turn on delay and turn off delay.

---

You can set the impulse and pause time for Clock and only one time for the other gates.

All times are millisecond values.

You can set the initial level for Clock.  
With set to HIGH the Clock pulse will start with HIGH, otherwise with LOW.

Note: Under OS2.0 the initial level gadgets are'nt disabled for gates other than Clock. Using this gadget won't effect anything in this case.

## 1.69 switchrequester

This requester appears when you click on a switch.

You can set the color of the switch when it is on/HIGH, the initial level and the type of the switch.

Per default a switch is turned off at start, with initial level set to HIGH the switch is turned on at start and you turn it off with first click.

With the type gadgets you can choose a real switch or a key.

If you use it as switch, you can click on it (in simulation mode) to turn it on (output=HIGH) and click again to turn it off (output=LOW).  
If you use it as key, clicking on it turns it on and releasing the button after this turns it off.

## 1.70 ledrequester

This requester appears when you click on a LED.

In this requester you can only set the color of the LED when it is on.

## 1.71 trafficlightrequester

---

This requester appears when you click on a traffic light.

Here you can select the screen-color for each light (red, yellow, green). Per default all colors are blue.

If you select "find best color", the program tries to allocate colors that match red, yellow and green. This feature is only available for OS3.0 and above.

You should use a workbench or custom screen with at least 8 colors for correct red, yellow and green color.

## 1.72 numdisplayrequester

This requester appears when you click on a numeric display.

Here you can set the mode for display (hex or decimal). You can also turn off the leading 0. For example, with leading 0 off the number 7 is shown as " 7", not as "07".

## 1.73 labelrequester

In the string-gadget you can input the label-text with max. 16 letters.

You can choose the text position above or below.

"Select font" opens a font requester where you can select a font for the label text. This font is used for ALL label texts, not just for this gate. The default font is TinyTxt.font supplied in the LogicSim archive.

## 1.74 author

If you want to write me or have questions, bug-reports or suggestions, please use one of the following addresses. I will reply on every mail.

email:           atetzl@hawk.sax.de

---

www: <http://www.sax.de/~atetzl>

snailmail: Andreas Tetzl  
Liebethaler Str. 18  
01796 Pirna  
GERMANY

You can find program-updates and new example circuits on my Homepage (<http://www.sax.de/~atetzl>).

## 1.75 credits

I must thank the following people, who helped me while developing LogicSim:

Jens Borsdorf  
for many good suggestion and beta testing

Falk Zuehlsdorff <[ai036@rz.tu-ilmenau.de](mailto:ai036@rz.tu-ilmenau.de)>  
for beta testing, suggestions and the PURITY

Christoph Feck <[c\\_feck@informatik.uni-kl.de](mailto:c_feck@informatik.uni-kl.de)>  
for the TinyTxt Font

Tino Bensing & Torsten May  
for the scanning of the about-picture

## 1.76 keyboard

These are only the shortcuts reachable without pressing the amiga key. You can find the shortcuts for menu items in the menus themselves.

gates  
-----

a - AND	A - NAND
o - OR	O - NOR
x - XOR	n - NOT
s - switch	l - LED
l - HIGH	0 - LOW

editing  
-----

---

c - connect	d - remove gate
r - Start (run)	w - remove wire
u - undo	2 - 2 inputs
3 - 3 inputs	5 - 5 inputs

SPACE - repeat last action

drawing wires (connect)

-----

SPACE - set knot to last point

u - remove last part of wire (undo last click)

ESC - abort drawing

simulation

-----

SPACE or p - pause simulation

ESC - abort simulation

## 1.77 hints

- Press space to repeat the last action.
  - To follow long wires, you can click on the wire to highlight it.
  - To 'debug' ~circuits, turn the 'highlight HIGH wire' item on. All wires with HIGH will be drawn white in simulation mode, so you can easily check if your circuit works as you want.
  - If you want to print your circuit, I recommend that you save it as iff and import it in your word processor. You will probably get a much better printing and you are able to include some text.
  - Click on a "basic gate" like AND, NAND, OR, NOR, XOR to replace it with another one.
-