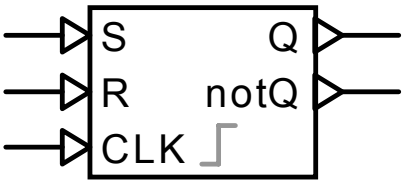
Flip-flop: S-R nand-gated rising-clock



S-R flip-flop
nand-gated

1 Description

This device is an implementation of a S-R nand-gated flip-flop with rising-edge clock and no override controls. For a version with the override controls, use the device " S-R nand-gated rising-clock full".

1.1 Pins

This device has five pins:

pin	*type*	*description*
S	input	S input
R	input	R input
CLK	input	rising-edge clock
Q	output	Q output
notQ	output	notQ output

1.2 Parameters

The initial value of *Q* must be defined if the device is possibly holding at *t=0*. When the device operates in clearing or setting mode at *t=0*, the initial value is ignored.

The value of the *stepped_mode* flag determines whether the device operates in *stepped* or *ramped* mode.
In *stepped* mode (the default for ideal logical signals), the outputs are represented as stepped signals, where changes in value are observed as vertical steps at the time they occur. In *ramped* mode, the value transitions of the outputs are seen as ramps between $t-\Delta t$ and *t*.

parameter	*description*
Q_ini	initial value of Q if holding or toggling at t=0
stepped_mode	=1 to indicate stepped mode (default)

	=0 to indicate ramped mode

1.3 Input

The input pins may be connected to any control signals.

Numerical input values are automatically interpreted as logical values by this device, as follows:

input	*converted logical value*	*logical value representation*
value > 0	true	1
value ≤ 0	false	0

1.4 Output

The outputs are *Q* and its logical inverse *not*Q. Their representation as *stepped* or *ramped* signals is determined by the value given to the parameter *stepped_mode*.

The numerical representation of the output logical values is:

output logical value	*output numerical value*
true	1
false	0

1.5 Representation

The implementation of the model can be inspected by opening the device's subcircuit.

The model applies the following logic for determining its state:

rule sequence	*action*	*output*
if not triggering else if S>0 and R>0 else if S<=0 and R<=0 else if S>0 else if S<=0 endif	holding setting holding setting clearing	Q(t) = Q(t-Δt) Q(t) = Q(t-Δt) Q(t) = Q(t-Δt) Q(t) = 1 Q(t) = 0
if holding at t=0	use Q_ini	Q(0) = Q_ini

where triggering occurs on a rising edge on the clock signal.