testbench.sv

SV/Verilog Testbench

```
// Code your testbench here
// or browse Examples
module jkff_test;
reg j,k, clk,reset;
wire q;

  jk_ff ob(.j(j),.k(k),.clk(clk),.reset(reset),.q(q)) ;

  initial begin
    clk= 0;
    forever #5 clk =~clk;
  end
  initial begin
    $dumpfile("test.vcd");
    $dumpvars(1);

  reset=1;
 j= 1; k= 1;
  #5 reset =0;
    repeat(10) @(negedge clk ) begin
     $display("j=%0h,k=%0h,q=%0h",j,k,q);
     j = $random;
     k = $random;
```

design.sv

SV/Verilog Design

```
// Code your design here
module jk_ff ( input j,input k,input clk,input reset ,output q);
   reg q;

  always @ (posedge clk )
    begin
    if(reset == 1)
      q <= 1'b0;
  else if (j==0 && k==0)
      q <= q;
  else if (j==1 && k==0)
    q <= 1'b1;
  else if (j==0 && k==1)
    q <= 1'b0;
  else
    q <= ~q;
    end
endmodule
```

Log | Share

```
VCD info: dumpfile test.vcd opened for output.
j=1,k=1,q=x
j=0,k=1,q=0
j=1,k=1,q=1
j=1,k=1,q=0
j=1,k=0,q=1
j=1,k=1,q=0
j=0,k=1,q=0
j=1,k=0,q=1
j=1,k=0,q=1
j=1,k=0,q=1
Finding VCD file...
```

EPWave

From: 0s To: 250s

Get Signals Radix 100%

Note: To revert to EPWave opening in a new browser window, set that option on your user page.

Languages & Libraries

Testbench + Design

VHDL

Libraries

None
OVL 2.8.1
OSVVM

Top entity

testbench

Enable VUnit

Tools & Simulators

Aldec Riviera Pro 2020.04

Compile Options

testbench.vhd

```vhdl
-- Testbench for OR gate
library IEEE;
use IEEE.std_logic_1164.all;

entity testbench is
-- empty
end testbench;

architecture tb of testbench is


component or_gate is
port(
  a: in std_logic;
  b: in std_logic;
  q: out std_logic);
end component;

signal a_in, b_in, q_out: std_logic;

begin


  DUT: or_gate port map(a_in, b_in, q_out);
```

design.vhd

VHDL Design

```vhdl
-- Simple OR gate design
library IEEE;
use IEEE.std_logic_1164.all;

entity or_gate is
port(
  a: in std_logic;
  b: in std_logic;
  q: out std_logic);
end or_gate;

architecture rtl of or_gate is
begin
  process(a, b) is
  begin
    q <= a or b;
  end process;
end rtl;

```

testbench.vhd

```
    wait for 1 ns;
    assert(q_out='1') report "Fail 0/1" severity error;

    a_in <= '1';
    b_in <= 'X';
    wait for 1 ns;
    assert(q_out='1') report "Fail 1/X" severity error;

    a_in <= '1';
    b_in <= '1';
    wait for 1 ns;
    assert(q_out='1') report "Fail 1/1" severity error;

    -- Clear inputs
    a_in <= '0';
    b_in <= '0';

    assert false report "Test done." severity note;
    wait;
  end process;
end tb;
```

design.vhd

VHDL Design

```
-- Simple OR gate design
library IEEE;
use IEEE.std_logic_1164.all;

entity or_gate is
port(
  a: in std_logic;
  b: in std_logic;
  q: out std_logic);
end or_gate;

architecture rtl of or_gate is
begin
  process(a, b) is
  begin
    q <= a or b;
  end process;
end rtl;
```

Log | Share

```
[2022-04-15 14:55:05 UTC] vlib work && vcom "-2008" design.vhd testbench.vhd  && vsim -c -do "vsim testbench; run -all; exit"
VSIMSA: Configuration file changed: `/home/runner/library.cfg'
ALIB: Library "work" attached.
work = /home/runner/work/work.lib
Aldec, Inc. VHDL Compiler, build 2020.04.130
VLM Initialized with path: "/home/runner/library.cfg".
DAGGEN WARNING DAGGEN_0523: "The source is compiled without the -dbg switch. Line breakpoints and assertion debug will not be available."
COMP96 File: design.vhd
COMP96 Compile Entity "or_gate"
COMP96 Compile Architecture "rtl" of Entity "or_gate"
COMP96 File: testbench.vhd
COMP96 Compile Entity "testbench"
```

Log Share

```
COMP96 Compile Entity "testbench"
COMP96 Compile Architecture "tb" of Entity "testbench"
COMP96 Compile success 0 Errors 0 Warnings  Analysis time :  30.0 [ms]
dmesg: read kernel buffer failed: Operation not permitted
dmesg: read kernel buffer failed: Operation not permitted
# Aldec, Inc. Riviera-PRO version 2020.04.130.7729 built for Linux64 on June 10, 2020.
# HDL, SystemC, and Assertions simulator, debugger, and design environment.
# (c) 1999-2020 Aldec, Inc. All rights reserved.
# ELBREAD: Elaboration process.
# ELBREAD: Elaboration time 0.0 [s].
# KERNEL: Main thread initiated.
# KERNEL: Kernel process initialization phase.
```

Log Share

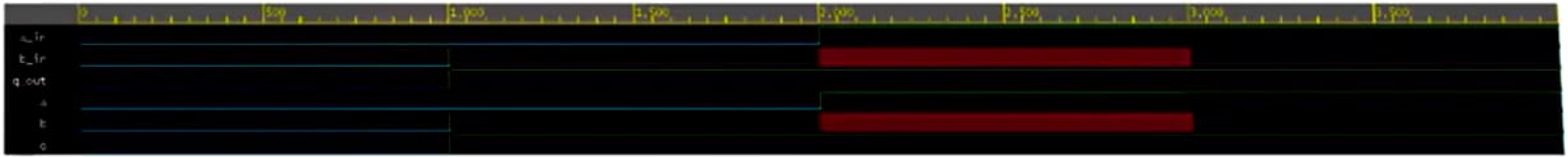
```
# ELAB2: You do not have a license to run VHDL performance optimized simulation. Contact Aldec for ordering information - sales@aldec.com.
# ELAB2: Elaboration final pass complete - time: 0.0 [s].
# KERNEL: Warning: You are using the Riviera-PRO EDU Edition. The performance of simulation is reduced.
# KERNEL: Warning: Contact Aldec for available upgrade options - sales@aldec.com.
# KERNEL: Kernel process initialization done.
# Allocation: Simulator allocated 5400 kB (elbread=427 elab2=4829 kernel=142 sdf=0)
# KERNEL: ASDB file was created in location /home/runner/dataset.asdb
# EXECUTION:: NOTE    : Test done.
# EXECUTION:: Time: 4 ns,  Iteration: 0,  Instance: /testbench,  Process: line__26.
# KERNEL: Simulation has finished. There are no more test vectors to simulate.
# VSIM: Simulation has finished.
```

Done

EPWave

From: 0ps To: 4,000ps

Get Signals Radix 100%



Note: To revert to EPWave opening in a new browser window, set that option on your user page.

void(0);

Languages & Libraries

Testbench + Design

VHDL

Libraries

None
OVL 2.8.1
OSVVM

Top entity

testbench

Enable VUnit

Tools & Simulators

Aldec Riviera Pro 2020.04

Compile Options

testbench.vhd

```
24  DUT: or_gate port map(a_in, b_in, q_out);
25
26  process
27  begin
28    a_in <= '0';
29    b_in <= '0';
30    wait for 1 ns;
31    assert(q_out='0') report "Fail 0/0" severity error;
32
33    a_in <= '0';
34    b_in <= '1';
35    wait for 1 ns;
36    assert(q_out='1') report "Fail 0/1" severity error;
37
38    a_in <= '1';
39    b_in <= 'X';
40    wait for 1 ns;
41    assert(q_out='1') report "Fail 1/X" severity error;
42
43    a_in <= '1';
44    b_in <= '1';
45    wait for 1 ns;
46    assert(q_out='1') report "Fail 1/1" severity error;
```

design.vhd

VHDL Design

```
1  -- Simple OR gate design
2  library IEEE;
3  use IEEE.std_logic_1164.all;
4
5  entity or_gate is
6  port(
7    a: in std_logic;
8    b: in std_logic;
9    q: out std_logic);
10 end or_gate;
11
12 architecture rtl of or_gate is
13 begin
14   process(a, b) is
15   begin
16     q <= a or b;
17   end process;
18 end rtl;
19
```