/\*

\* RS-trigger with assynch. reset

\*/

library ieee;

use ieee.std\_logic\_1164.all;

entity RS\_trigger is

generic (T: Time := 0ns);

port ( R, S : in std\_logic;

Q, nQ : out std\_logic;

reset, clock : in std\_logic );

end RS\_trigger;

architecture behaviour of RS\_trigger is

signal QT: std\_logic; -- Q(t)

begin

process(clock, reset) is

subtype RS is std\_logic\_vector (1 downto 0);

begin

if reset = '0' then

QT <= '0';

else

if rising\_edge(C) then

if not (R'stable(T) and S'stable(T)) then

QT <= 'X';

else

case RS'(R&S) is

when "01" => QT <= '1';

when "10" => QT <= '0';

when "11" => QT <= 'X';

when others => null;

end case;

end if;

end if;

end if;

end process;

Q <= QT;

nQ <= not QT;

end architecture behaviour;