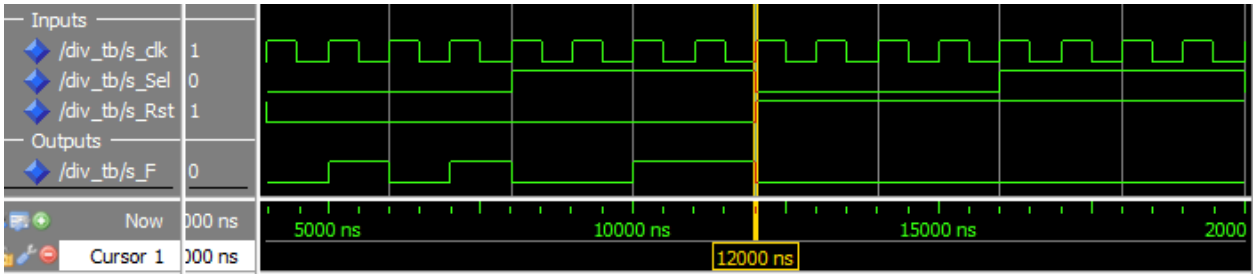
```vhdl
library IEEE;
use IEEE.std_logic_1164.all;

entity DFF_sync is
   port(i_D       : in std_logic;
            i_Clk : in std_logic;
            i_Rst : in std_logic;

            o_Q   : out std_logic);
end DFF_sync;

architecture behavior of DFF_sync is
begin
      process(i_Clk, i_Rst)
      begin

            -- We are now asynchronously resettable
            if(i_Rst = '1') then
                  o_Q <= '0';
            elsif(rising_edge(i_Clk)) then
                  o_Q <= i_D;
            end if;

      end process;
end behavior;
```

```vhdl
library IEEE;
use IEEE.std_logic_1164.all;
use ieee.numeric_std.all;

entity Mux2to1Modded is
      port(i_A  : in std_logic;
             i_B  : in std_logic;
             i_S  : in std_logic;
             o_F  : out std_logic);

end Mux2to1Modded;

architecture dataflow of Mux2to1Modded is
begin
      o_F <= i_A when (i_S = '0') else i_B;
      --o_F <= (i_A and (not i_S)) or (i_B and i_S);

end dataflow;
```

```vhdl
library IEEE;
use IEEE.std_logic_1164.all;
use ieee.numeric_std.all;

entity ClockDivider is
      port(i_Clk  : in std_logic;
            i_Rst      : in std_logic;
            i_Sel      : in std_logic;

            o_F  : out std_logic);

end ClockDivider;


architecture structure of ClockDivider is

      component DFF_sync
         port(i_D             : in std_logic;
                  i_Clk       : in std_logic;
                  i_Rst       : in std_logic;
                  o_Q               : out std_logic);
      end component;

      component Mux2to1Modded
         port(i_A          : in std_logic;
                  i_B               : in std_logic;
                  i_S               : in std_logic;
                  o_F         : out std_logic);
      end component;

      signal s_D1, s_D2       : std_logic;
      signal s_notD1, s_notD2 : std_logic;


begin

      DFF1: DFF_sync
      port map(   i_D         => s_notD1,
                        i_Clk => i_Clk,
                        i_Rst => i_Rst,
                        o_Q         => s_D1);

      s_notD1 <= not s_D1;

      DFF2: DFF_sync
      port map(   i_D         => s_notD2,
```

```
                        i_Clk => s_notD1,
                        i_Rst => i_Rst,
                        o_Q         => s_D2);

      s_notD2 <= not s_D2;

      mux: Mux2to1Modded
      port map(   i_A         => s_D1,
                        i_B         => s_D2,
                        i_S         => i_Sel,
                        o_F         => o_F);

end structure;
```



```
library IEEE;
use IEEE.std_logic_1164.all;

-- 1Mhz => 1 us

entity Div_tb is
end Div_tb;

architecture behavior of Div_tb is
      signal s_clk            : std_logic := '1';

      signal s_Sel      : std_logic;
      signal s_Rst      : std_logic;
      signal s_F        : std_logic;
begin

DUT : entity work.ClockDivider
  port map (i_Clk   => s_clk,
```

```
                  i_Rst => s_Rst,
                  i_Sel => s_Sel,

                  o_F     => s_F);

      s_clk <= not s_clk after 500 ns;

process is
begin

      s_Rst   <= '1';
      s_Sel <= '0';
    wait for 4 us;
      -- Reset everything
      -- Wait for four clock pulses

      s_Rst   <= '0';
      s_Sel <= '0';
    wait for 4 us;
      -- Wait for four clock pulses

    s_Rst   <= '0';
      s_Sel <= '1';
    wait for 4 us;
      -- Wait for four clock pulses

      s_Rst   <= '1';
      s_Sel <= '0';
    wait for 4 us;
      -- Wait for four clock pulses

    s_Rst   <= '1';
      s_Sel <= '1';
    wait for 4 us;
      -- Wait for four clock pulses

end process;
end behavior;
```