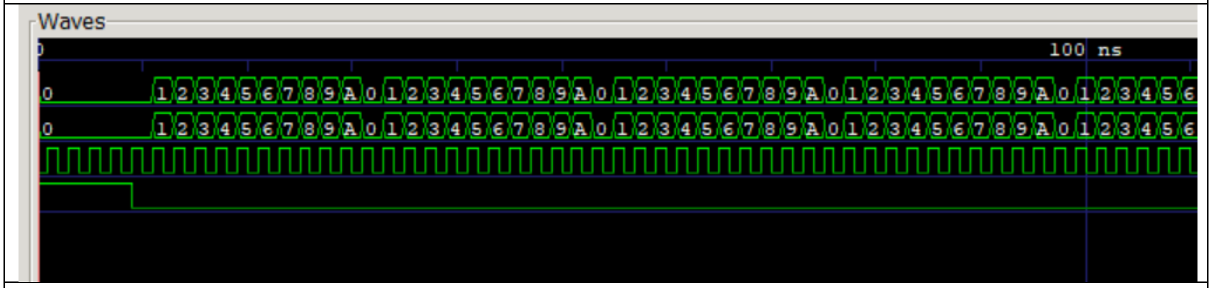
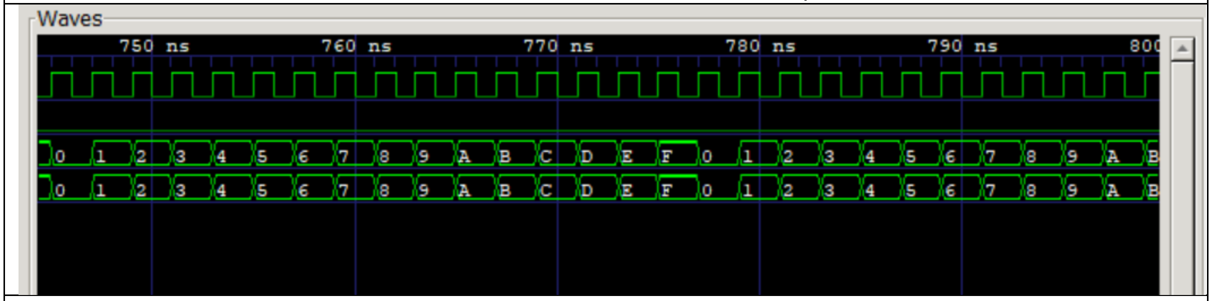
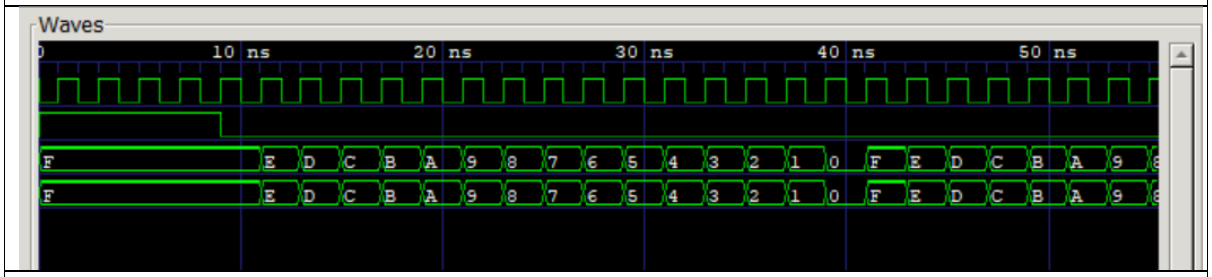
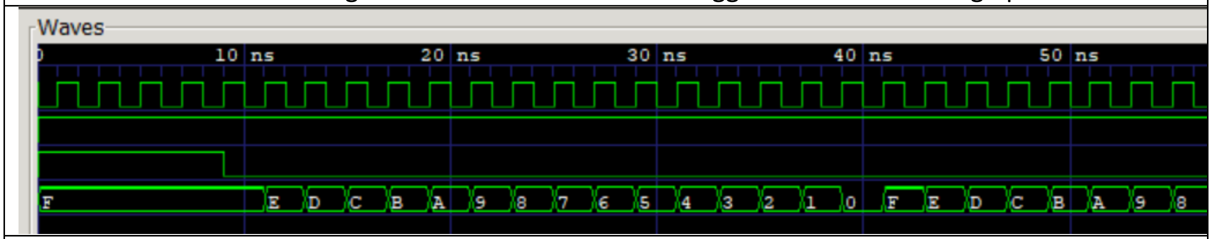
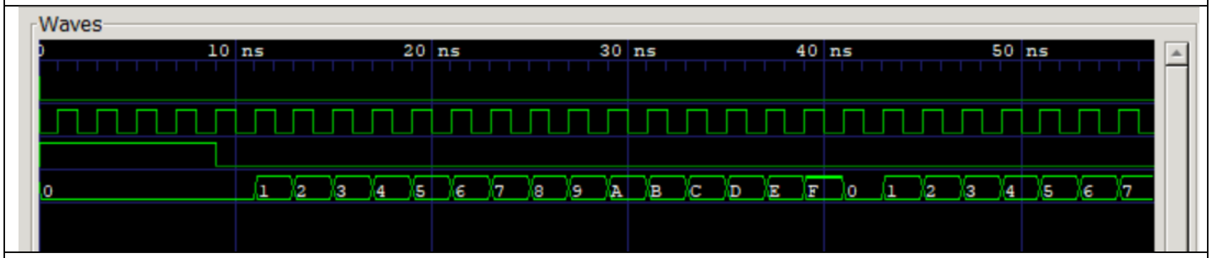
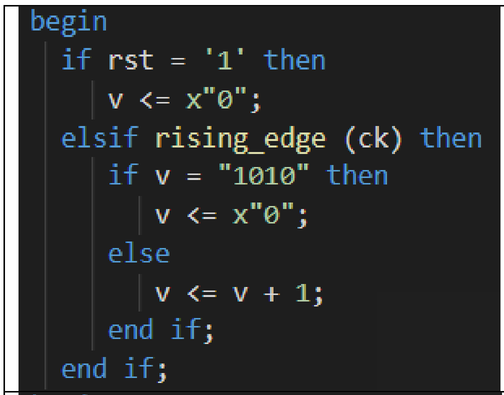
Computer Systems Practical 6 25 May 2021

Walt Deyzel: 21750793

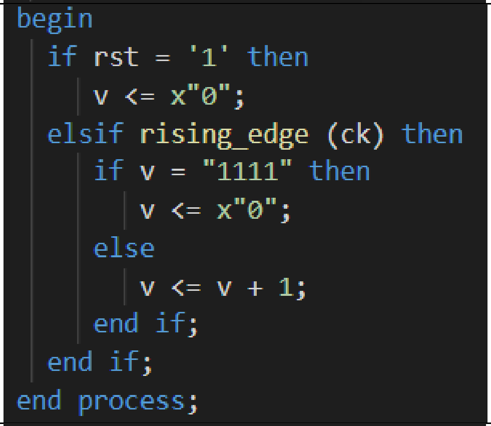
Counter results

Modified code counter results Count up to F

Modified code counter results. Count down from F

With an if statement looking at the state if DEC one can toggle between counting up or down.

If dec == 0




```vhdl
begin
  if rst = '1' then
    v <= x"0";
  elsif rising_edge (ck) then
    if v = "1010" then
      v <= x"0";
    else
      v <= v + 1;
    end if;
  end if;
```

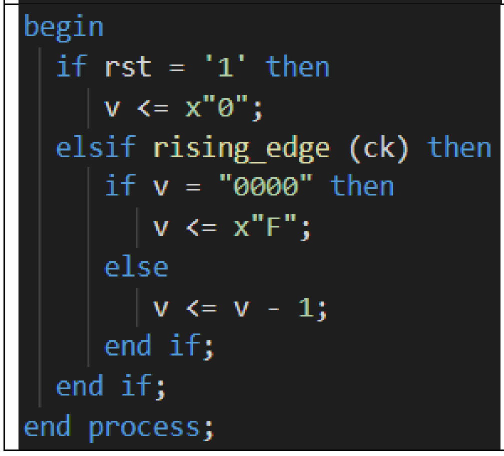
The counter counts till ten then is set to zero as seen in the code on the left. Ten in binary is 1010.

The code is then modified to count to F. F is 15 in decimal and 1111 in binary.



```vhdl
begin
  if rst = '1' then
    v <= x"0";
  elsif rising_edge (ck) then
    if v = "1111" then
      v <= x"0";
    else
      v <= v + 1;
    end if;
  end if;
end process;
```

Modified code to allow counter to count to 15.

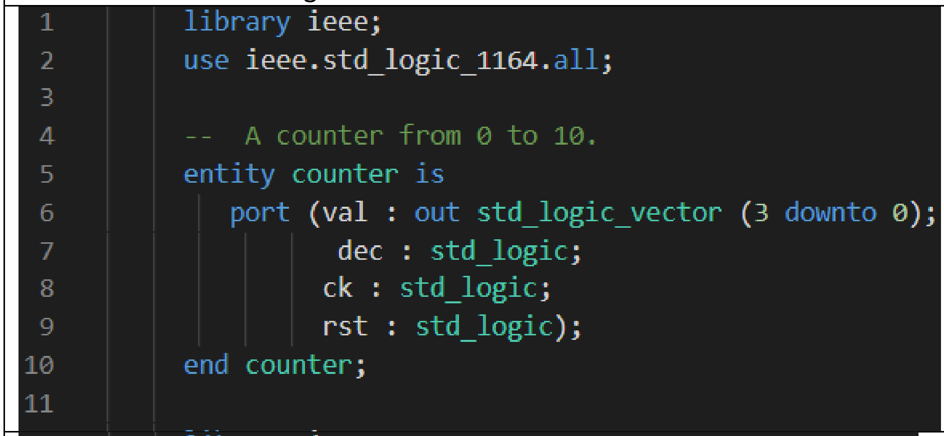


```vhdl
begin
  if rst = '1' then
    v <= x"0";
  elsif rising_edge (ck) then
    if v = "0000" then
      v <= x"F";
    else
      v <= v - 1;
    end if;
  end if;
end process;
```

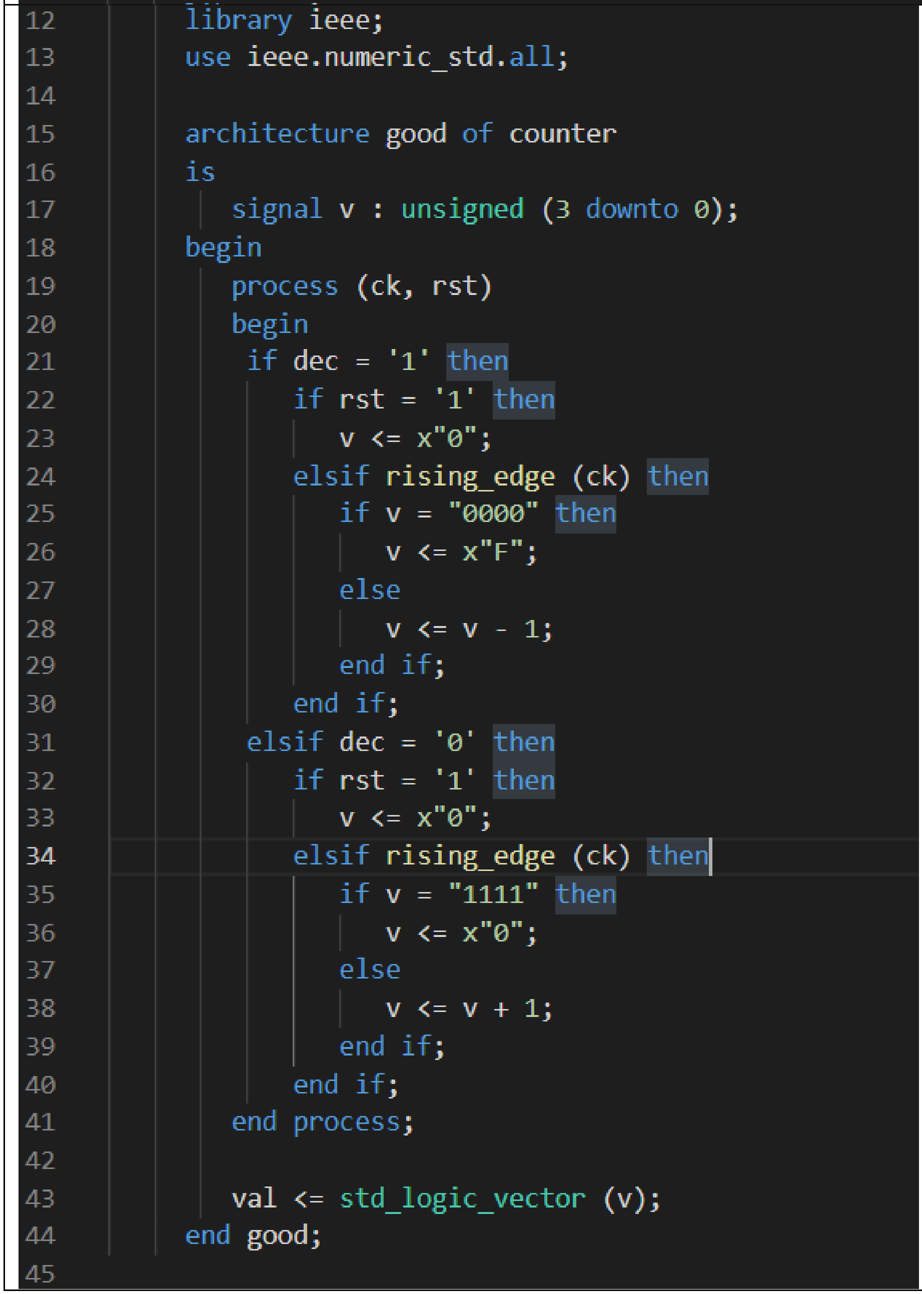
Modified code to count down from F to zero and then repeat.

Adding a if statement allows one to toggle between using a up counter or a down counter.

Counter code with changes

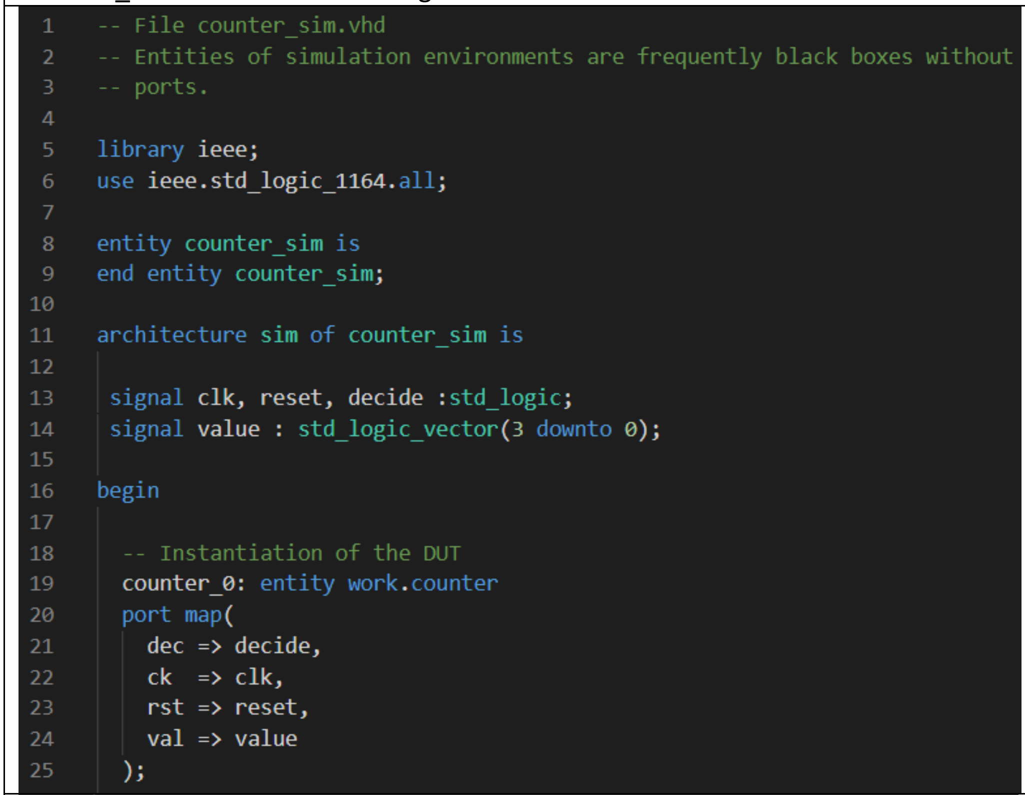


```
library ieee;
use ieee.std_logic_1164.all;

--  A counter from 0 to 10.
entity counter is
   port (val : out std_logic_vector (3 downto 0);
          dec : std_logic;
         ck : std_logic;
         rst : std_logic);
end counter;

```

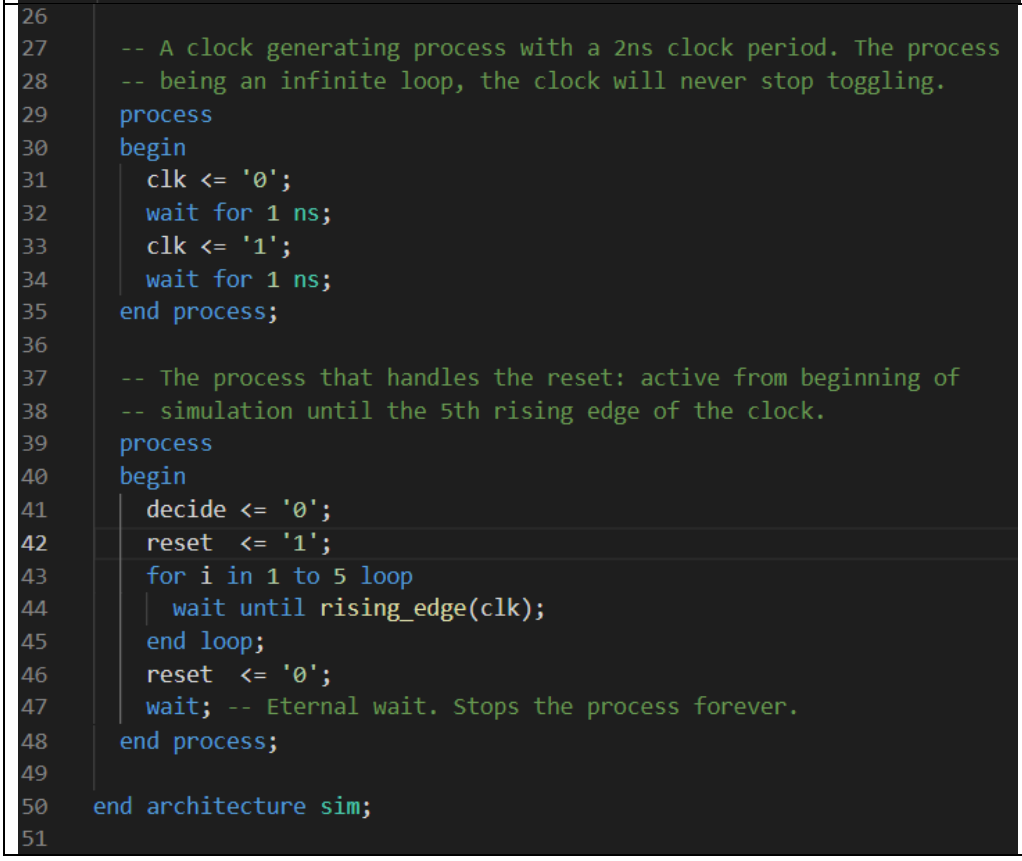


```
library ieee;
use ieee.numeric_std.all;

architecture good of counter
is
   signal v : unsigned (3 downto 0);
begin
   process (ck, rst)
   begin
    if dec = '1' then
      if rst = '1' then
         v <= x"0";
      elsif rising_edge (ck) then
         if v = "0000" then
            v <= x"F";
         else
            v <= v - 1;
         end if;
      end if;
    elsif dec = '0' then
      if rst = '1' then
         v <= x"0";
      elsif rising_edge (ck) then
         if v = "1111" then
            v <= x"0";
         else
            v <= v + 1;
         end if;
      end if;
   end process;

   val <= std_logic_vector (v);
end good;

```

Counter_sim.vhdl with code changes



```vhdl
-- File counter_sim.vhd
-- Entities of simulation environments are frequently black boxes without
-- ports.

library ieee;
use ieee.std_logic_1164.all;

entity counter_sim is
end entity counter_sim;

architecture sim of counter_sim is

 signal clk, reset, decide :std_logic;
 signal value : std_logic_vector(3 downto 0);

begin

  -- Instantiation of the DUT
  counter_0: entity work.counter
  port map(
    dec => decide,
    ck  => clk,
    rst => reset,
    val => value
  );
```



```vhdl

  -- A clock generating process with a 2ns clock period. The process
  -- being an infinite loop, the clock will never stop toggling.
  process
  begin
    clk <= '0';
    wait for 1 ns;
    clk <= '1';
    wait for 1 ns;
  end process;

  -- The process that handles the reset: active from beginning of
  -- simulation until the 5th rising edge of the clock.
  process
  begin
    decide <= '0';
    reset  <= '1';
    for i in 1 to 5 loop
      wait until rising_edge(clk);
    end loop;
    reset  <= '0';
    wait; -- Eternal wait. Stops the process forever.
  end process;

end architecture sim;

```