```
--------------------------------------------------------------------------------
--
--  Purpose:
--
--    Synchronous Counter
--
--  Discussion:
--
--
--  Licensing:
--
--    This code is distributed under the GNU LGPL license.
--
--  Modified:
--
--    2012.03.28
--
--  Author:
--
--    Young W. Lim
--
--  Parameters:
--
--    Input:
--
--    Output:
--------------------------------------------------------------------------------

library STD;
use STD.textio.all;

library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;


entity counter is
  port (
    clk, rst,    : in   std_logic := '0';
    en           : in   std_logic := '0';
    dq           : out  std_logic_vector (4 downto 0) := X"00");

end counter;

architecture rtl of counter is

  component adder
    port (an, bn,    : in  std_logic_vector (4 downto 0) := X"00";
          ci,        : in  std_logic := '0';
          cn,        : out  std_logic_vector (4 downto 0) := X"00";
          co         : out  std_logic := '0');
  end component;

  signal dinInc : std_logic_vector := 4 downto 0;

begin

  inc: adder(an=>din, bn=>X"00", ci=>'1', cn=>dinInc, co=>);

  Reg: process (clk, rst)
  begin  -- process Reg
    if rst = '0' then                   -- asynchronous reset (active low)
      dq <= X"00";
    elsif clk'event and clk = '1' then  -- rising clock edge
      if (en='1') then
        dq <= dinInc;
      end if;
    end if;
  end process Reg;
```

```
end rtl
```