Code

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


entity counter is
port ( --   En                :      in      std_logic;
         data_in              :      in      std_logic_vector(7 downto 0);
         clk                  :      in      std_logic;
         read_fifo            :      in      std_logic;
         wr_fifo              :      in      std_logic;
         data_out             :      out     std_logic_vector(7 downto 0)
         );
end entity;

architecture behavior of counter is
component fifo_1 is
port
   (
      clock    : IN STD_LOGIC ;
      data     : IN STD_LOGIC_VECTOR (7 DOWNTO 0);
      rdreq    : IN STD_LOGIC ;
      wrreq    : IN STD_LOGIC ;
      q        : OUT STD_LOGIC_VECTOR (7 DOWNTO 0)
   );
END component;

begin
U1 : fifo_1
port map(
            data     =>    data_in,
            clock    =>    clk,
            rdreq    =>    read_fifo,
            wrreq    =>    wr_fifo,
            q        =>     data_out );

end architecture;
```

TESTBENCH

```
begin

UUT: counter
port map (
                                --En                    =>          En_tb,
                                data_in         =>      data_in_tb,
                                clk                     =>          clk_tb,
                                read_fifo       =>      read_fifo_tb,
                                wr_fifo         =>      wr_fifo_tb,
                                data_out                =>      data_out_tb );


process
begin
                        clk_tb  <=      '0';
                        wait for 10 ns;
                        clk_tb  <=      '1';
                        wait for 10 ns;
end process;

process
begin
        wr_fifo_tb <= '0';
                        wait for 45 ns;
                        wr_fifo_tb <= '1';
                        wait for 50 ns;
                        wr_fifo_tb <= '0';
                        wait;
end process;

read_fifo       : process
begin
        read_fifo_tb <= '0';
                        wait for 150 ns;
                        read_fifo_tb <= '1';
                        wait for 50 ns;
                        wr_fifo_tb <= '0';
                        wait;
end process;

data_in : process
begin
        data_in_tb <= "00000000";
```

```
                    wait for 30 ns;
                    data_in_tb <= "01010101";
                    wait for 50 ns;
                    data_in_tb <= "00000000";
                    wait;
end process;
end architecture;
```

Waveform