Solution: TP4

Part I:

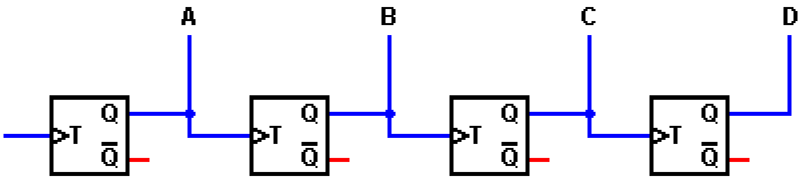


Figure 1. A 4-bit counter.

```
LIBRARY ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
ENTITY lab4_1 IS
PORT ( l: in std_logic;
              clk: in std_logic;
              r: in std_ulogic;
              c: out std_logic_vector(3 downto 0)
             );
end lab4_1;
architecture behavior of lab4_1 is
signal n: std_logic_vector (3 downto 0) := (others => '0');
begin
             process(clk,r)
             begin
        if (rising_edge(clk)) then
             if(r ='1') then n <="0000";
                           if (l='1') then
                                  if (n<= "1111") then n <= "0000";
                                  else n <= n+1;
                                  end if;
                                  end if;
                     end if;
             end if;
             c<=n;
             end process;

end behavior;
```

16-bit counter

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_arith.all;
entity lab4_2 is
port ( clk :in std_logic;
            c : out std_logic_vector(15 downto 0));
end lab4_2;

architecture behaviore of lab4_2 is

signal cmp: std_logic_vector(15 downto 0);
begin
process(clk)
begin
if ( rising_edge(clk)) then
                  cmp <=cmp+1;
             end if;
      end process;
c<=cmp;
end behaviore;
```

Part II:

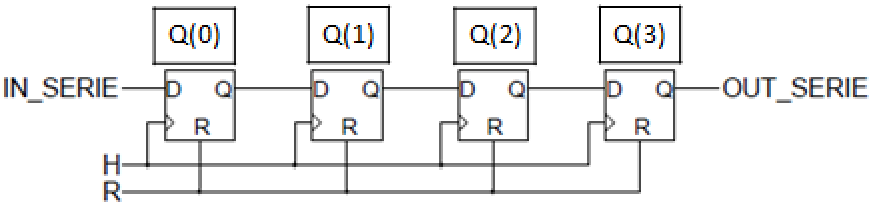
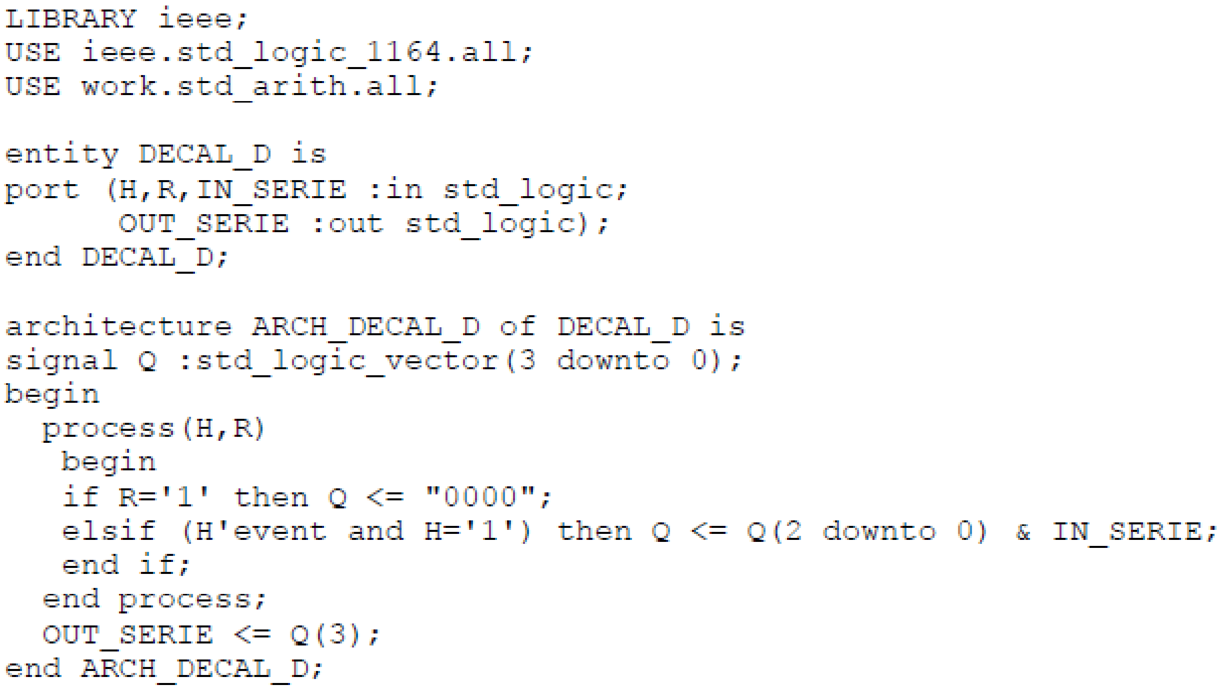


Figure 2. Shift register serial in –serial out.

Shift register serial in –serial out:

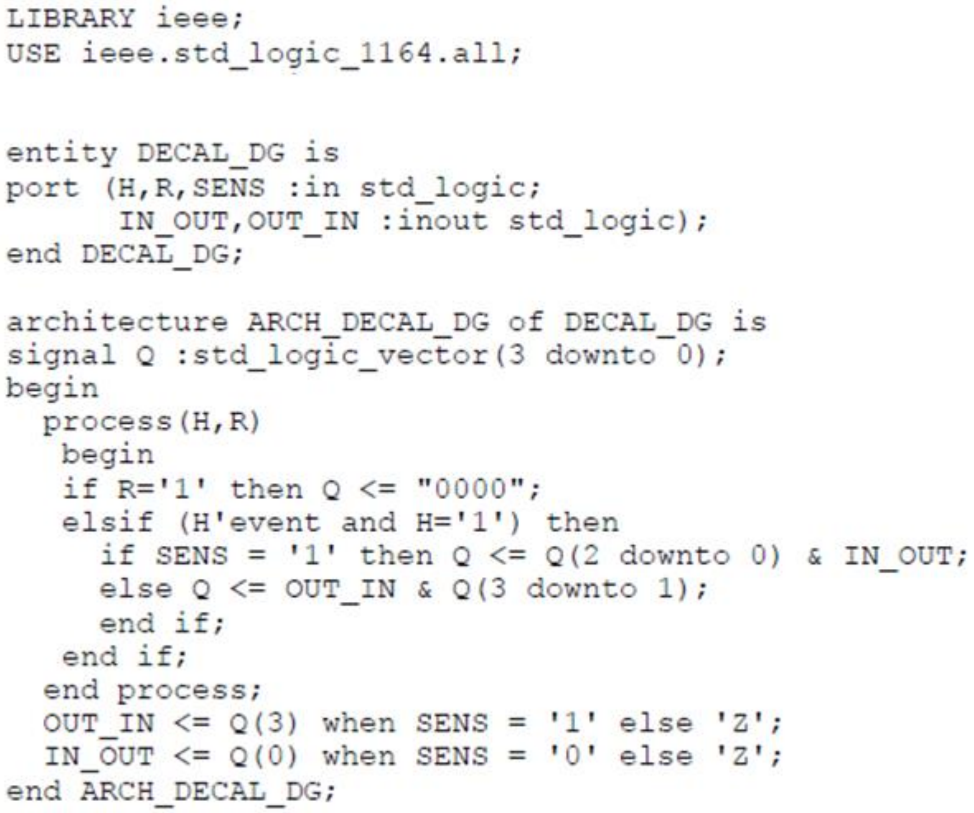


```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE work.std_arith.all;

entity DECAL_D is
port (H,R,IN_SERIE :in std_logic;
      OUT_SERIE :out std_logic);
end DECAL_D;

architecture ARCH_DECAL_D of DECAL_D is
signal Q :std_logic_vector(3 downto 0);
begin
  process(H,R)
   begin
   if R='1' then Q <= "0000";
   elsif (H'event and H='1') then Q <= Q(2 downto 0) & IN_SERIE;
   end if;
  end process;
  OUT_SERIE <= Q(3);
end ARCH_DECAL_D;
```

Shift register left or right:



```
LIBRARY ieee;
USE ieee.std_logic_1164.all;


entity DECAL_DG is
port (H,R,SENS :in std_logic;
      IN_OUT,OUT_IN :inout std_logic);
end DECAL_DG;

architecture ARCH_DECAL_DG of DECAL_DG is
signal Q :std_logic_vector(3 downto 0);
begin
  process(H,R)
   begin
   if R='1' then Q <= "0000";
   elsif (H'event and H='1') then
     if SENS = '1' then Q <= Q(2 downto 0) & IN_OUT;
     else Q <= OUT_IN & Q(3 downto 1);
     end if;
   end if;
  end process;
  OUT_IN <= Q(3) when SENS = '1' else 'Z';
  IN_OUT <= Q(0) when SENS = '0' else 'Z';
end ARCH_DECAL_DG;
```