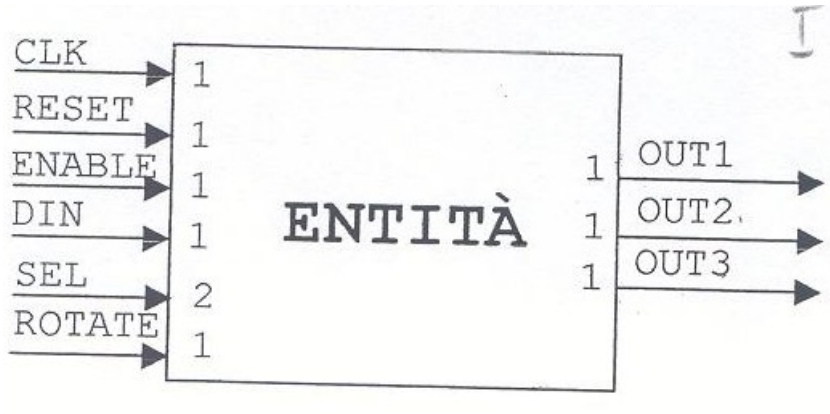
19 September 2008

Create a clock-synchronous VHDL entity which splits input serial data (**DIN**) into 3 serial output streams.

Starting from a reset, a selection 2-bits signal (**SEL**) is sampled. This signal determines which output starts to stream while the other outputs stay at high impedance.

The input signal **ROTATE** determines a change in the active output following an order decided by the student.

The **ENABLE** signal deactivates the outputs and keep them at high impedance while the entity internal function is not affected.



```
1   library IEEE;
2   use IEEE.STD_LOGIC_1164.ALL;
3   use IEEE.STD_LOGIC_ARITH.ALL;
4   use IEEE.STD_LOGIC_UNSIGNED.ALL;
5
6   entity selection is
7      port (CLK, RST, EN, DIN, ROT: in std_logic;
8            SEL: in std_logic_vector (1 downto 0);
9            OUT1, OUT2, OUT3: out std_logic);
10  end selection;
11
12  architecture Behavioral of selection is
13
14     signal REG: std_logic_vector (1 downto 0):="00";
15     signal reg1, reg2, reg3: std_logic:='Z';
16
17     begin
18
19        process (CLK, RST)
20        begin
21           if (RST = '1') then
22              REG <= SEL;
23              OUT1 <= 'Z';
24              OUT2 <= 'Z';
25              OUT3 <= 'Z';
26           elsif rising_edge (CLK) then
27              if (REG = "00") then
28                 REG <= SEL;
29                 OUT1 <= 'Z';
30                 OUT2 <= 'Z';
31                 OUT3 <= 'Z';
32              elsif (REG = "01") then
33                 reg1 <= DIN;
34                 reg2 <= 'Z';
35                 reg3 <= 'Z';
36                    if (ROT = '1') then
37                       REG <= "10";
38                    end if;
39              elsif (REG = "10") then
40                 reg1 <= 'Z';
41                 reg2 <= DIN;
42                 reg3 <= 'Z';
43                 if (ROT = '1') then
44                    REG <= "11";
45                 end if;
46              elsif (REG = "11") then
47                 reg1 <= 'Z';
48                 reg2 <= 'Z';
49                 reg3 <= DIN;
50                 if (ROT = '1') then
51                    REG <= "01";
52                 end if;
53              end if;
54
55           end if;
56        end process;
57
58  OUT1 <= reg1 when (EN = '1') else 'Z';
59  OUT2 <= reg2 when (EN = '1') else 'Z';
60  OUT3 <= reg3 when (EN = '1') else 'Z';
61
62  end Behavioral;
```