19 September 2013

Starting from an activation signal **EN**, the entity must produce a serial output **DOUT** a sample of the 4-bits parallel input **DIN** (`b3b2b1b0`) determined by a parallel 2-bits input signal **SEL**. **SEL** determines the index of the last bit to transfer to the output.

Example:
`SEL` = 10 = 2 → Output sequence = `b0b1b2`

When the output sequence is concluded, a new one is computed depending on the values of **DIN** and **SEL** at that moment.

DOUT must be in high impedance state when the entity is disabled and for a clock period between the end of an output sequence and the next one.

The serial output **PULSE** must normally be equal to 0; it switches to 1 to signal the end of a **DOUT** sequence during the last clock period of the sequence sequence (in the previous example, when `b2` is transferred to **DOUT**).

Note that while **DOUT** is transferring a sequence **DIN** and **SEL** can assume other values but the entity must ignore them until the end to the output sequence.

```
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 P_to_S is
7      port (CLK, EN: in std_logic;
8            DIN: in std_logic_vector (3 downto 0);
9            SEL: in std_logic_vector(1 downto 0);
10           DOUT, PULSE: out std_logic);
11  end P_to_S;
12
13  architecture Behavioral of P_to_S is
14
15     signal cnt, index: integer range 0 to 10:=0;
16     signal regin: std_logic_vector(3 downto 0):="0000";
17     signal regout: std_logic:='Z';
18
19     begin
20
21        process (CLK, EN)
22        begin
23
24           if rising_edge(CLK) then
25              if (cnt = 0) then
26                 regin <= DIN;
27                 regout <= 'Z';
28                 index <= conv_integer(SEL);
29                 cnt <= cnt + 1;
30              elsif ((cnt > 0) and (cnt < (index + 1))) then
31                 regout <= regin (cnt - 1);
32                 cnt <= cnt + 1;
33              elsif ((cnt > 0) and (cnt = (index + 1))) then
34                 regout <= regin (cnt - 1);
35                 cnt <= 0;
36              end if;
37           end if;
38
39        end process;
40
41  DOUT <= regout when (EN = '1') else 'Z';
42  PULSE <= '1' when (cnt = 0) else '0';
43
44  end Behavioral;
```