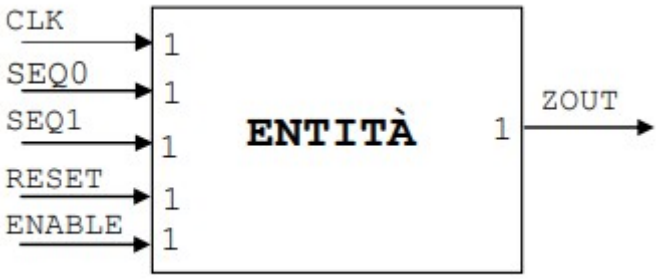
16 June 2009

Create a clock-synchronous VHDL entity which, starting from a reset, analyses the values of 2 serial inputs **SEQ0** and **SEQ1**. The output **ZOUT** must be 0 when the number of bits equal to 0 coming through **SEQ0** is 3 units higher than that coming through **SEQ1**. If the condition is reversed, **ZOUT** must be equal to 1. In any other case and when the entity is not enabled, **ZOUT** must be in high impedance.



```
1   --  Possibile soluzione della prova del 16/06/2010- C. Giaconia   --
2
3   library IEEE;
4   use IEEE.STD_LOGIC_1164.ALL; use IEEE.STD_LOGIC_ARITH.ALL; use IEEE.STD_LOGIC_UNSIGNED.ALL;
5
6   entity SOL20100616 is
7   port(clk,reset,enable : in std_logic;
8      seq0,seq1 : in std_logic;
9      zout : out std_logic);
10  end SOL20100616;
11
12  architecture Behavioral of SOL20100616 is
13
14  signal count0,count1 : integer;
15  -- contatori per il numero di zeri di seq0 e seq1
16
17  begin
18
19  process(clk,reset)
20  begin
21    if reset = '1' then               -- al reset azzero i contatori
22          count0 <= 0;  count1 <= 0;
23    elsif (clk'event and clk = '1') then
24         -- qua conto gli 0 da seq0
25         if seq0 = '0' then  count0 <= count0 + 1; end if;
26         -- qua conto gli 0 da seq1
27         if seq1 = '0' then count1 <= count1 + 1; end if;
28    end if;
29  end process;
30
31  zout <=
32      -- zout pari a 0 se count0 supera di 3 count1
33      '0' when (count0 > count1 + 3 and enable = '0') else
34      -- zout pari a 1 se count1 supera di 3 count0
35      '1' when (count1 > count0 + 3 and enable = '0') else
36      -- zout pari a Z in tutti gli altri casi
37      'Z';
38  end Behavioral;
39
40
```