```vhdl
--------------------------------------------------------------------------------
-- HEIG-VD, Haute Ecole d'Ingenierie et de Gestion du canton de Vaud
-- Institut REDS, Reconfigurable & Embedded Digital Systems
--
-- Fichier      : enc_prio_16in_top
-- Description  : Encodeur de priorite à 16 entrees
--
-- Auteur       : E. Messerli
-- Date         : 30.09.2009
-- Version      : 0.0
--
-- Utilise      : Labo systeme numerique
--
--| Modifications |------------------------------------------------------------
-- Ver  Auteur  Date       Description
-- 1.0   GAA  31.08.2016   Adaptation fichier vhdl pour utilisation avec Quartus
-- 1.1   EMI  05.10.2016   Adaptation pour encodeur a 16 entress
--                         entity: utilise vecteurs pour in_i et num_o
--
--------------------------------------------------------------------------------

library ieee;
use ieee.std_logic_1164.ALL;
--use ieee.numeric_std.ALL;

entity enc_prio_16in_top is
   port(
      in_i     : in     std_logic_vector(15 downto 0);
      detect_o : out    std_logic;
      num_o    : out    std_logic_vector(3 downto 0)
   );
end enc_prio_16in_top ;

architecture struct of enc_prio_16in_top is

  -- Component Declarations
  component enc_prio_4in
    port(
    en_i     : in     std_logic;
    in0_i    : in     std_logic;
    in1_i    : in     std_logic;
    in2_i    : in     std_logic;
    in3_i    : in     std_logic;
    detect_o : out    std_logic;
    en_o     : out    std_logic;
    num0_o   : out    std_logic;
    num1_o   : out    std_logic
    );
  end component;

  for all : enc_prio_4in use entity work.enc_prio_4in(flot_don);

  -- Internal Declarations
  constant ENABLE      : std_logic:='1';

  signal vect_detect_s, vect_enable_s : std_logic_vector(3 downto 0);
  signal vect_num0_s, vect_num1_s : std_logic_vector(3 downto 0);

begin

  ENC4_3 : enc_prio_4in port map (
      en_i => ENABLE,
      in0_i => in_i(12)  ,
      in1_i => in_i(13)  ,
      in2_i => in_i(14)  ,
      in3_i => in_i(15)  ,
      detect_o => vect_detect_s(3) ,
      en_o  => vect_enable_s(3) ,
      num0_o =>  vect_num0_s(3) ,
```

```
70          num1_o  => vect_num1_s(3)
71      );
72
73      ENC4_2 : enc_prio_4in port map (
74          en_i => vect_enable_s(3),
75          in0_i => in_i(8)  ,
76          in1_i => in_i(9)  ,
77          in2_i => in_i(10)  ,
78          in3_i => in_i(11)  ,
79          detect_o => vect_detect_s(2) ,
80          en_o  => vect_enable_s(2) ,
81          num0_o =>  vect_num0_s(2) ,
82          num1_o  => vect_num1_s(2)
83      );
84
85      ENC4_1 : enc_prio_4in port map (
86          en_i => vect_enable_s(2),
87          in0_i => in_i(4)  ,
88          in1_i => in_i(5)  ,
89          in2_i => in_i(6)  ,
90          in3_i => in_i(7)  ,
91          detect_o => vect_detect_s(1) ,
92          en_o  => vect_enable_s(1) ,
93          num0_o =>  vect_num0_s(1) ,
94          num1_o  => vect_num1_s(1)
95      );
96
97      ENC4_0 : enc_prio_4in port map (
98          en_i => vect_enable_s(1),
99          in0_i => in_i(0)  ,
100         in1_i => in_i(1)  ,
101         in2_i => in_i(2)  ,
102         in3_i => in_i(3)  ,
103         detect_o => vect_detect_s(0) ,
104         en_o  => vect_enable_s(0) ,
105         num0_o =>  vect_num0_s(0) ,
106         num1_o  => vect_num1_s(0)
107     );
108
109     num_o(3) <= vect_detect_s(3) or vect_detect_s(2);
110     num_o(2) <= vect_detect_s(3) or vect_detect_s(1);
111     num_o(1) <= vect_num1_s(3) or vect_num1_s(2) or vect_num1_s(1) or vect_num1_s(0);
112     num_o(0) <= vect_num0_s(3) or vect_num0_s(2) or vect_num0_s(1) or vect_num0_s(0);
113     detect_o <= vect_detect_s(3) or vect_detect_s(2) or vect_detect_s(1) or
        vect_detect_s(0);
114
115 end struct;
116
```