```
 1: library STD;
 2: library IEEE;
 3: use IEEE.std_logic_1164.all;
 4:
 5: entity Compare is
 6:     port(
 7:         in1     :   in  std_logic_vector(2 downto 0);
 8:         in2     :   in  std_logic_vector(2 downto 0);
 9:         out1    :   out std_logic
10:     );
11: end Compare;
12:
13: architecture structural of Compare is
14:
15:     component invX1
16:     port(
17:         in1     :   in  std_logic;
18:         out1    :   out std_logic
19:     );
20:     end component;
21:
22:     component nand2
23:     port(
24:         in1     :   in  std_logic;
25:         in2     :   in  std_logic;
26:         out1    :   out std_logic
27:     );
28:     end component;
29:
30:     component nor2
31:     port(
32:         in1     :   in  std_logic;
33:         in2     :   in  std_logic;
34:         out1    :   out std_logic
35:     );
36:     end component;
37:
38:     component xnor2
39:     port(
40:         in1     :   in  std_logic;
41:         in2     :   in  std_logic;
42:         out1    :   out std_logic
43:     );
44:     end component;
45:
46:     signal xor_out  :   std_logic_vector(2 downto 0);
47:     signal nor_out  :   std_logic;
48:     signal inv      :   std_logic;
49:
50:     for xor_0, xor_1, xor_2 :   xnor2 use entity work.xnor2(structural);
51:     for nor_0 : nand2 use entity work.nand2(structural);
52:     for nor_1 :  nor2 use entity work.nor2(structural);
53:     for i   :   invX1 use entity work.invX1(structural);
54:
55: begin
56:
57:     xor_0   :   xnor2    port map(in1(0), in2(0), xor_out(0));
58:     xor_1   :   xnor2    port map(in1(1), in2(1), xor_out(1));
59:     xor_2   :   xnor2    port map(in1(2), in2(2), xor_out(2));
60:     nor_0   :   nand2    port map(xor_out(0), xor_out(1), nor_out);
61:     i       :   invX1   port map(xor_out(2), inv);
62:     nor_1   :   nor2    port map(inv, nor_out, out1);
63:
64: end structural;
```