```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity bubble is
  port (S1 : in std_logic;
        south_b : in std_logic;
        north_b : in std_logic;
        data_out : out std_logic_vector(7 downto 0));
       end bubble;


      architecture bubble_sort of bubble is
        type memory is array (1 to 5) of std_logic_vector(7 downto 0);

        begin

        SORT : process(S1,south_b,north_b)

         variable rami : memory :=
("00000001","00000010","00000100","00001000","00010000");
             --rami(1) := "00000100" ;
             --rami(2) := "00010000" ;
             --rami(3) := "00000100" ;
             -- rami(4) := "00100000" ;
             --rami(5) := "10000000" ;
         constant ramo : memory := rami ;
         variable t1,t2: integer;
         variable tmp : std_logic_vector(7 downto 0);
         variable x : integer := 1;
         variable y : integer := 1;
         begin

           for i in 1 to 5 loop
             t1 := 6;
             while t1 > i loop
             t2 := t1-2;
             t1 := t1-1;
            if(t1 >= 2) then
              if rami(t1) < rami(t2) then
                   tmp := rami(t1);
                   rami(t1) := rami(t2);
                   rami(t2) := tmp;
                   end if;
                   end if;
```

```
            end loop;
           end loop;

    if (S1 = '1') and (south_b = '1') then
       data_out <= rami(5);
       x := 4;
    elsif (S1 = '1') and (north_b = '1' )then
      data_out <= rami(x) ;
       x := x-1;
        if (x = 0) then
          x := 5;
        end if;
     end if;


   if (S1 = '0') and (south_b = '1') then
       data_out <= ramo(1);
       y := 2;
    elsif (S1 = '0') and (north_b = '1' )then
      data_out <= ramo(x) ;
       y := y+1;
        if (y = 6) then
          y := 1;
        end if;
    end if;


end process SORT;
end bubble_sort;
```