```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;
use IEEE.std_logic_unsigned.all;

entity full_adder is
      Port(a: in std_logic; b: in std_logic; c: in std_logic; sum: out
std_logic; carry: out std_logic);
end full_adder;

architecture Behaviorial of full_adder is
begin
      sum <= a xor b xor c;
      carry <= (a and b) or (b and c) or (c and a);
end Behaviorial;


library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;
use IEEE.std_logic_unsigned.all;

entity adder_4bit is
      Port(a: in std_logic_vector(3 downto 0); b: in std_logic_vector(3
downto 0); s: out std_logic_vector(3 downto 0); c: out std_logic);
end adder_4bit;

architecture Structural of adder_4bit is
Component full_adder is
      Port(a: in std_logic; b: in std_logic; c: in std_logic; sum: out
std_logic; carry: out std_logic);
end component;
Signal ic: std_logic_vector(3 downto 1);
begin
      fa1: full_adder port map(a(0),b(0),'0',s(0),ic(1));
      fa2: full_adder port map(a(1),b(1),ic(1),s(1),ic(2));
      fa3: full_adder port map(a(2),b(2),ic(2),s(2),ic(3));
      fa4: full_adder port map(a(3),b(3),ic(3),s(3),c);
end Structural;
```