## 4 bit Adder VHDL models

```
Tuesday, March 23, 2021
                     4:14 PM
library IEEE;
    IEEE.STD LOGIC 1164.all;
    IEEE.NUMERIC STD.all;
entity adder4 is
      port (op0: in std logic vector(3 downto 0);
             op1: in std logic vector(3 downto 0);
            cin: in std logic;
                  out std logic vector(3 downto 0);
            cout: out std logic);
end adder4;
-- Several architectures. Only the last one is synthetized.
architecture Structural of adder4 is
signal i carry: std logic vector(2 downto 0);
begin
                                                                  Master Time Bar: 0 ps
      FA0: entity work.FullAdder(DataFlow)
                                                                                 0 ps
                                                                                     80.0 ns
                                                                           Value at
              port map (ai => op0(0),
                                                                     Name
                                                                            0 ps
                         bi => op1(0),
                                                                          B 1
                                     ci => cin,
                                                                                 0010 X 1111 X 1100 X 1000 X 0011
                                                                          B 0010
                                     sum => s(0),
                                                                  🟲 > op1
                                                                          B 1010
                                                                                 <u> 1010 X 0111 X 0101 X 0010 X 0111 X 0001 X 1111 X 0011 X 1111 X 1011 X 0100 X 0110 X</u>
                                     co => i carry(0));
                                                                                 1101 X 0111 X 0001 X 1010 X 1011 X 1001 X 0010 X 1101 X 0010 X 1000 X 1001
                                                                  ⇒ > s
                                                                          B 1101
      FA1: entity work.FullAdder(DataFlow)
                                                                          B<sub>0</sub>
                                                                     cout
              port map (ai => op0(1),
                         bi => op1(1),
                                     ci => i carry(0),
                                     sum => s(1),
                                     co => i carry(1));
      FA2: entity work.FullAdder(DataFlow)
              port map (ai => op0(2),
                         bi => op1(2),
                                     ci => i carry(1),
                                     sum => s(2),
                                     co => i carry(2));
      FA3: entity work.FullAdder(DataFlow)
              port map (ai => op0(3),
                         bi => op1(3),
                                     ci => i carry(2),
                                     sum => s(3),
                                     co => cout);
end Structural;
architecture DataFlow of adder4 is
signal i carry: std logic vector(2 downto 0);
begin
             s(0) \le op0(0) xor op1(0) xor cin;
  i carry(^{\circ}) <= (op0(^{\circ}) and op1(^{\circ})) or (cin and (op0(^{\circ}) or op1(^{\circ})));
          s(1) \le op0(1) xor op1(1) xor i carry(0);
  i carry(1) \leq (op0(1) and op1(1)) or (i carry(0) and (op0(1) or op1(1)));
          s(2) \le op0(2) xor op1(2) xor i carry(1);
  i carry(^{2}) <= (op0(^{2}) and op1(^{2})) or (i carry(^{1}) and (op0(^{2}) or op1(^{2})));
          s(3) \le op0(3) xor op1(3) xor i carry(2);
         cout \leq (op0(3) and op1(3)) or (i carry(2) and (op0(3) or op1(3)));
end DataFlow;
architecture Arithm of Adder4 is
signal s operand0, s operand1, s result : unsigned(4 downto 0);
begin
    s operand0 <= '0' & unsigned(op0);
    s operand1 <= '0' & unsigned(op1);
    s result <= s operand0 + s operand1 + 1 when cin ='1' else
                     s operand0 + s operand1;
              s <= std logic vector(s result(3 downto 0));</pre>
          cout <= s result(4);</pre>
end Arithm;
```

**Example Simulation** 

Pointer: 329.6 ns

160<u>.</u>0 ns

240,0 ns

320,0 ns

1001

Interval: 329.6 ns

480,0 ns

560,0 ns

0011

400,0 ns