**--Full adder using half adder**

**1. Code**

----------------------------------------------------------------------------------

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity FULL\_ADDER\_STR is

Port ( a,b,c : in STD\_LOGIC;

sum,carry : out STD\_LOGIC);

end FULL\_ADDER\_STR;

architecture Structural of FULL\_ADDER\_STR is

component halfadd

port (a,b: in STD\_LOGIC ;

sum,carry : OUT STD\_LOGIC );

end component ;

SIGNAL sum1,carry1, sum2, carry2 : STD\_LOGIC ;

begin

halfadder1: halfadd port map (a => a, b => b, sum => sum1 ,carry => carry1 );

halfadder2: halfadd port map (a => sum1, b => c, sum =>sum ,carry =>carry2 );

carry <= carry1 or carry2 ;

end structural;

**Code 2: for Half Adder**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity halfadd is

Port ( a,b, : in STD\_LOGIC;

sum,carry : out STD\_LOGIC);

end halfadd;

architecture dataflow of halfadd is

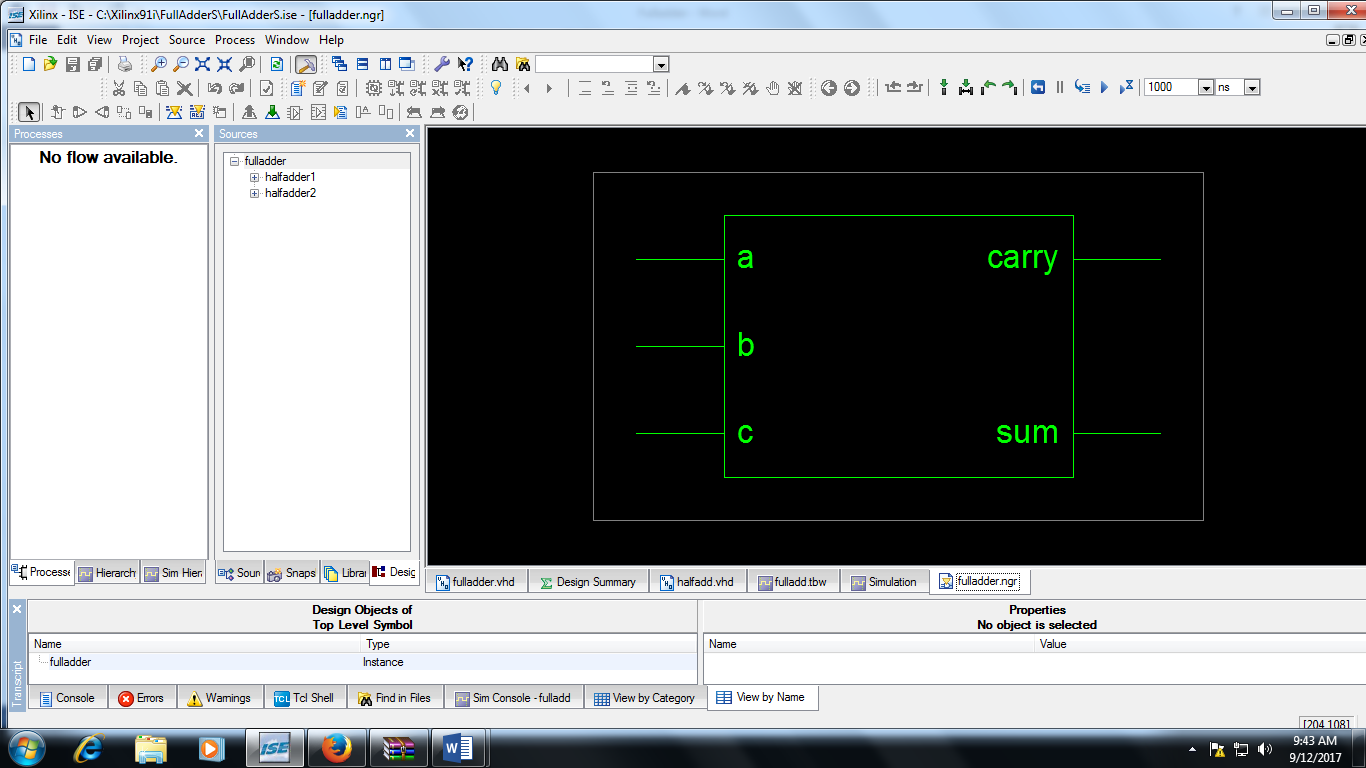
begin

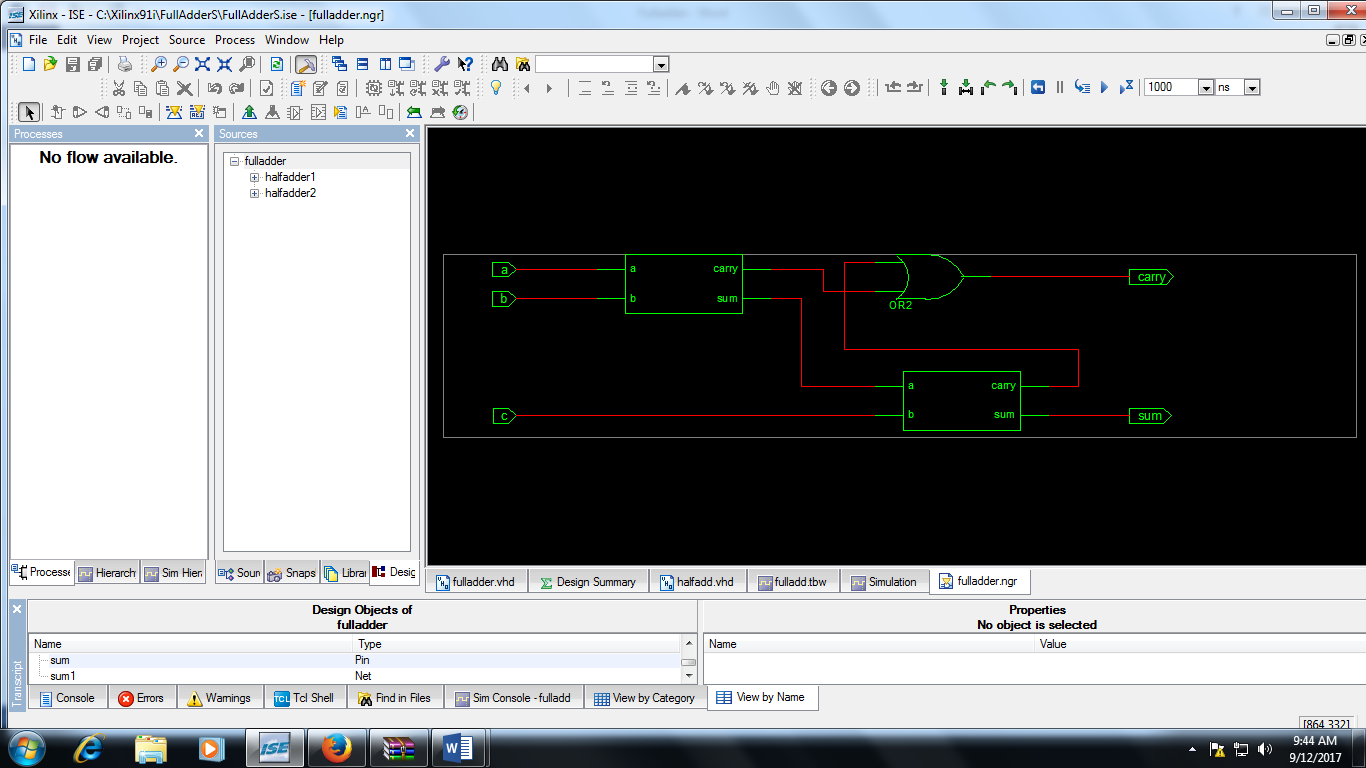
sum <= a xor b;

carry <= a and b;

end dataflow;

**2. RTL Schematic**





**3. Simulation Output**

