**Problem Statement :- Implement Full Adder using Structural Modelling**

**Structural Modeling for Full Adder**

First implement Half Adder :-

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

-- Company:

-- Engineer:

--

-- Create Date: 15:50:53 09/24/2015

-- Design Name:

-- Module Name: HA - Behavioral

-- Project Name:

-- Target Devices:

-- Tool versions:

-- Description:

--

-- Dependencies:

--

-- Revision:

-- Revision 0.01 - File Created

-- Additional Comments:

--

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

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

---- Uncomment the following library declaration if instantiating

---- any Xilinx primitives in this code.

--library UNISIM;

--use UNISIM.VComponents.all;

entity HA is

Port ( a : in STD\_LOGIC;

b : in STD\_LOGIC;

sum : out STD\_LOGIC;

carry : out STD\_LOGIC);

end HA;

architecture Behavioral of HA is

begin

sum<= a xor b;

carry<= a and b;

end Behavioral;

Now implement Full Adder:-

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

-- Company:

-- Engineer:

--

-- Create Date: 15:53:04 09/24/2015

-- Design Name:

-- Module Name: FA - Behavioral

-- Project Name:

-- Target Devices:

-- Tool versions:

-- Description:

--

-- Dependencies:

--

-- Revision:

-- Revision 0.01 - File Created

-- Additional Comments:

--

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

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

---- Uncomment the following library declaration if instantiating

---- any Xilinx primitives in this code.

--library UNISIM;

--use UNISIM.VComponents.all;

entity FA is

Port ( a : in STD\_LOGIC;

b : in STD\_LOGIC;

c : in STD\_LOGIC;

sum : out STD\_LOGIC;

carry : out STD\_LOGIC);

end FA;

architecture Behavioral of FA is

component HA is

Port ( a : in STD\_LOGIC;

b : in STD\_LOGIC;

sum : out STD\_LOGIC;

carry : out STD\_LOGIC);

end component;

signal sum1,carry1,carry2 : std\_logic;

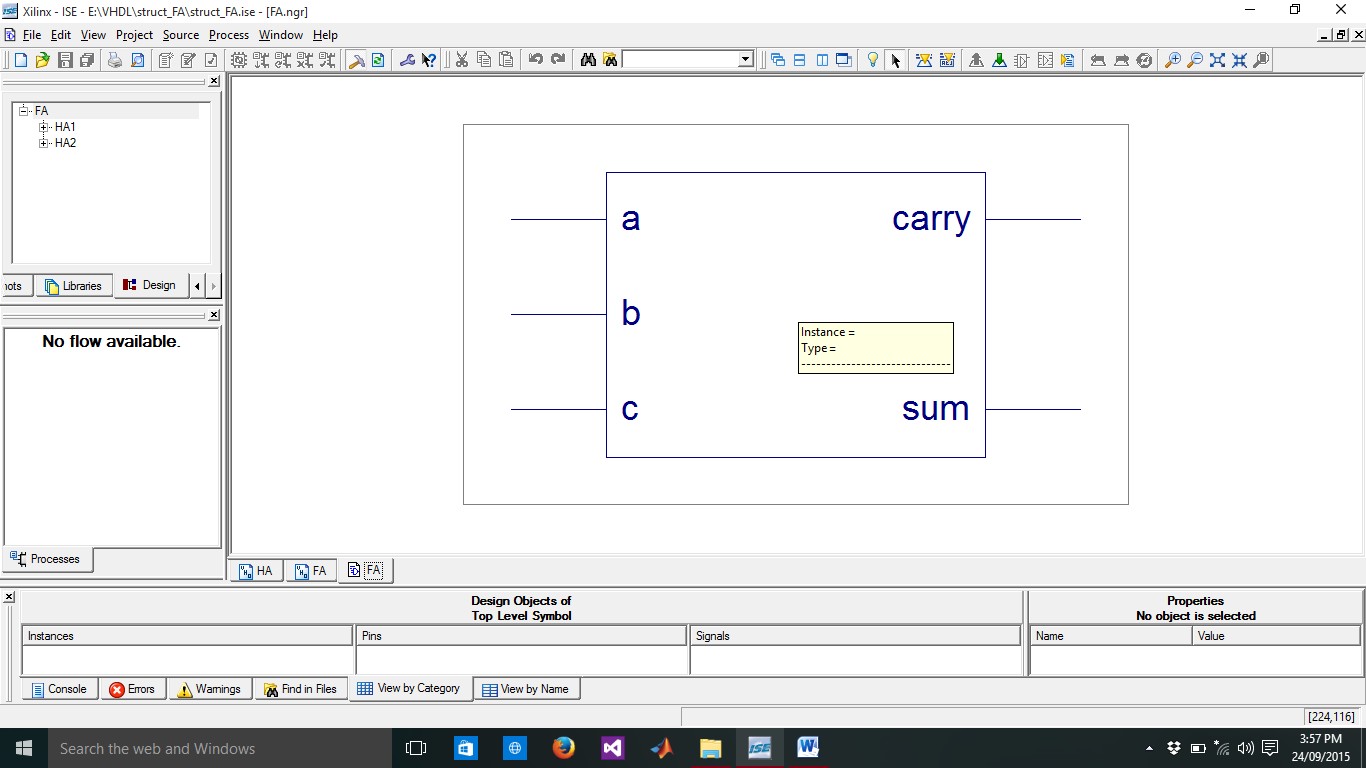
begin

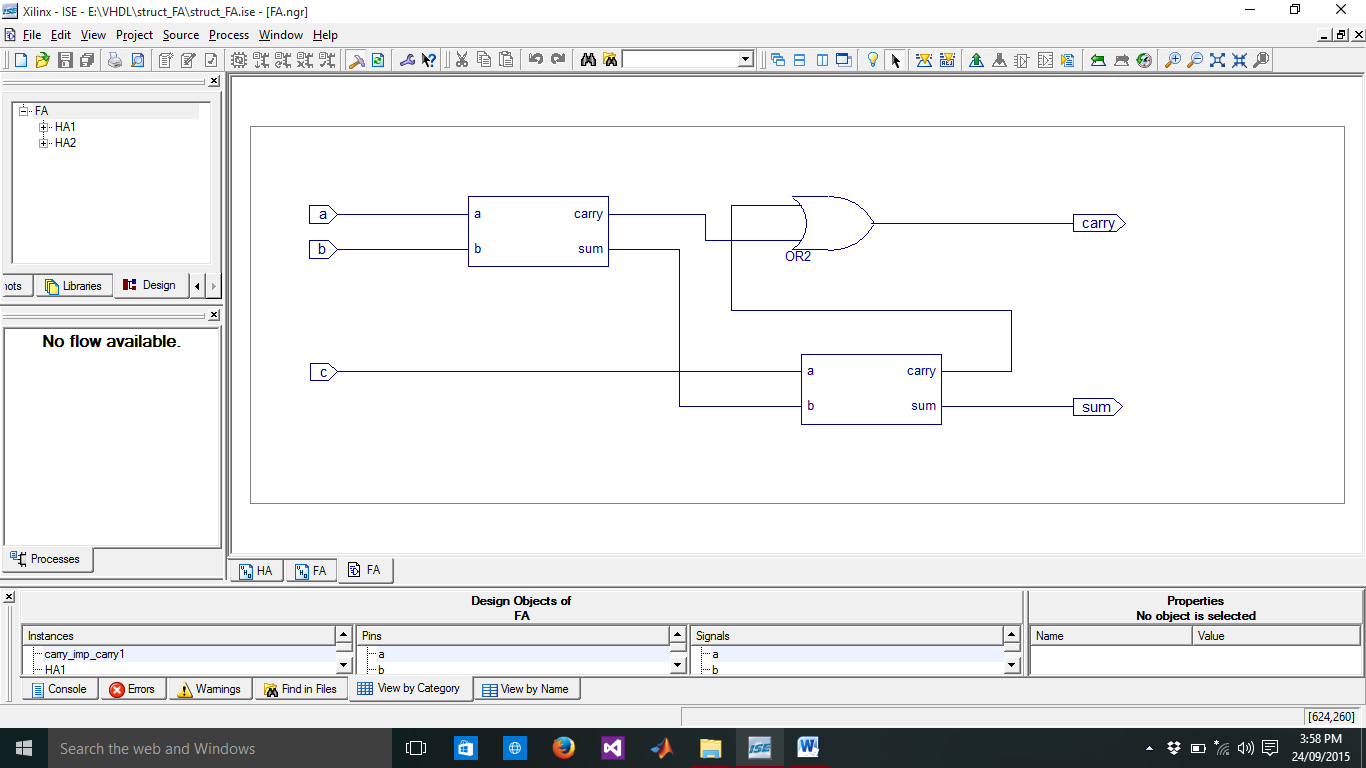
HA1 : ha port map (a,b,sum1,carry1);

HA2 : ha port map (c,sum1,sum,carry2);

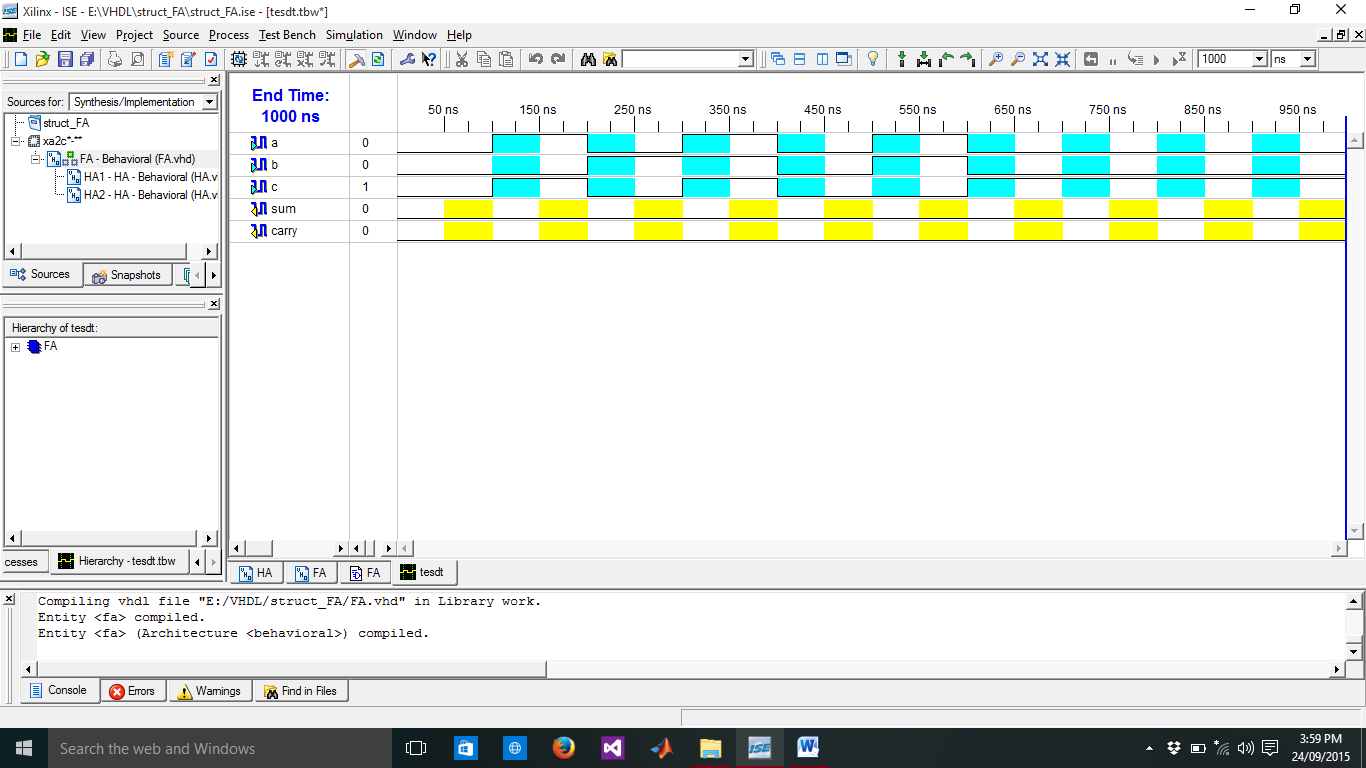
carry<= carry1 or carry2;

end Behavioral;





Input Waveform:-



Output Waveform:-

