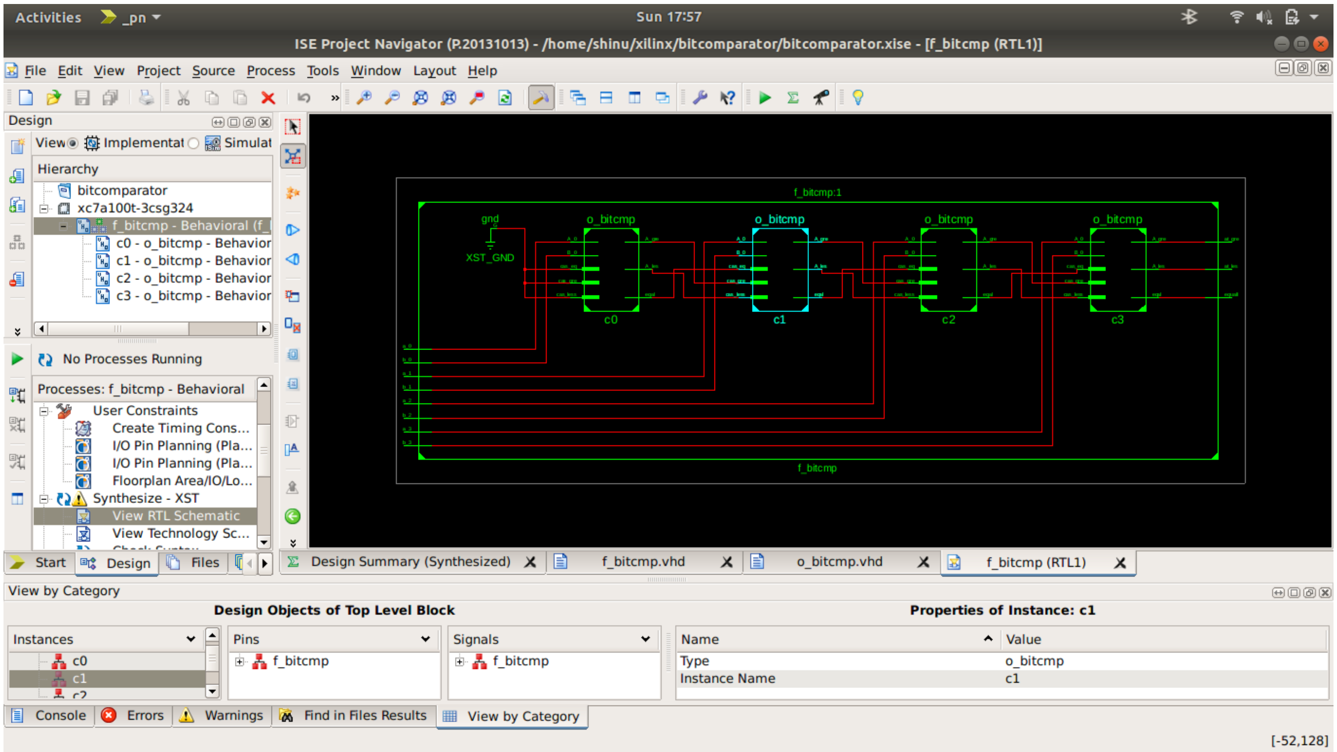
VHDL ASSIGNMENT
SHINU SHAJI C0761203

OBJECTIVE : DESIGN 4 BIT MAGNITUDE COMPARATOR USING STRUCTURAL APPROACH

SCHEMATICS :



ISE Project Navigator (P.20131013) - /home/shinu/xilinx/bitcomparator/bitcomparator.xise - [f_bitcmp (RTL1)]

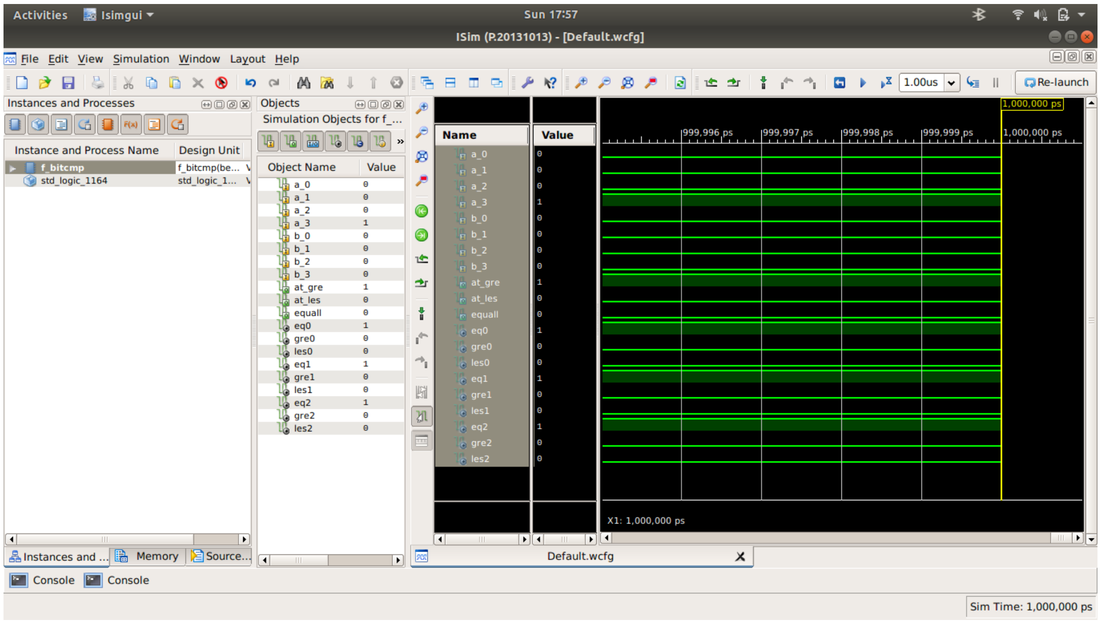
Design Objects of Top Level Block

Instances	Pins	Signals
c0	f_bitcmp	f_bitcmp
c1		
c2		

Properties of Instance: c1

Name	Value
Type	o_bitcmp
Instance Name	c1

PULSE WAVE FORM :



ISim (P.20131013) - [Default.wcfg]

Instance and Process Name	Design Unit
f_bitcmp	f_bitcmp(be...
std_logic_1164	std_logic_1...

Object Name	Value
a_0	0
a_1	0
a_2	0
a_3	1
b_0	0
b_1	0
b_2	0
b_3	0
at_gre	1
at_les	0
equall	0
eq0	1
gre0	0
les0	0
eq1	1
gre1	0
les1	0
eq2	1
gre2	0
les2	0

X1: 1,000,000 ps

Sim Time: 1,000,000 ps

SOURCE CODE :

ONE BIT COMPARATOR :

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date:    10:47:48 11/10/2019
-- Design Name:
-- Module Name:    o_bitcmp - 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;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponenrtts.all;

entity o_bitcmp is
port(
A_0 : in std_logic;
B_0 : in std_logic;
A_gre : out std_logic;
eqal : out std_logic;
A_les : out std_logic;
cas_gre : in std_logic;
cas_eq : in std_logic;
cas_less : in std_logic
);

end o_bitcmp;
```

```
architecture Behavioral of o_bitcmp is

signal eq:std_logic ;
signal gre:std_logic ;
signal less:std_logic ;
signal test : std_logic;

begin


gre <= A_0 and not(B_0);
less <= not(A_0) and B_0 ;
eq  <= (not(A_0) and not(B_0)) or (A_0 and B_0);
test <= cas_gre xor cas_eq  xor cas_less;
process(eq,less,gre,cas_gre,cas_eq,cas_less)
begin

if eq = '1'  and test ='1' then
        A_gre<=cas_gre;
        eqal<=cas_eq;
        A_les<=cas_less;
else
        A_gre<=gre;
        eqal<=eq;
        A_les<=less;

        end if;
end process;
end Behavioral;
```

4 BIT COMPARATOR :

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date:    11:05:50 11/10/2019
-- Design Name:
-- Module Name:    f_bitcmp - 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;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity f_bitcmp is
port(
        a_0 : in std_logic:='0';
        a_1 : in std_logic:='0';
        a_2 : in std_logic:='0';
        a_3 : in std_logic:='1';
        b_0 : in std_logic:='0';
        b_1 : in std_logic:='0';
        b_2 : in std_logic:='0';
        b_3 : in std_logic:='0';
        at_gre :out std_logic;
        at_les :out std_logic;
        equall :out std_logic
);
end f_bitcmp;

architecture Behavioral of f_bitcmp is
signal eq0 :std_logic;
signal gre0 : std_logic;
signal les0 : std_logic;
signal eq1 :std_logic;
signal gre1 : std_logic;
signal les1 : std_logic;
signal eq2 :std_logic;
signal gre2 : std_logic;
signal les2 : std_logic;
component o_bitcmp
port(
A_0   : in std_logic;
B_0   : in std_logic;
cas_gre : in std_logic;
cas_eq : in std_logic;
cas_less : in std_logic;
A_gre : out std_logic;
eqal  : out std_logic;
A_les : out std_logic);

end component ;

begin
c0:o_bitcmp port map (a_0,b_0,'0','0','0',gre0,eq0,les0);
c1:o_bitcmp port map (a_1,b_1,gre0,eq0,les0,gre1,eq1,les1);
c2:o_bitcmp port map (a_2,b_2,gre1,eq1,les1,gre2,eq2,les2);
c3:o_bitcmp port map (a_3,b_3,gre2,eq2,les2,at_gre,equall,at_les);
end Behavioral;
```