```vhdl
----------------------------------------------------------------------------------
-- Engineer: Olasubomi Borishade
--
-- Create Date: 02/04/2026 01:18:11 PM
-- Design Name: Comparison Logic Block
-- Module Name: Comparison_Logic_Block - Behavioral
-- Project Name: EENG 5560 Homework 2

-- Description: Testing a logic block capable of performing 6 types of comparisons:
-- greater than, less than, equal to, greater than or equal to, less than or equal
to and not equal to
-- on two inputs A and B

-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use work.OPERATIONS_ARRAY_CUSTOM_PACK.ALL;
use IEEE.NUMERIC_STD.ALL;
use STD.ENV.FINISH;

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

entity Comparison_Logic_Block_Test_Bench is
--   Port ( );
end Comparison_Logic_Block_Test_Bench;

architecture Behavioral of Comparison_Logic_Block_Test_Bench is
Constant d_w_c: integer := 3; -- Declaring data width signal
Signal As, Bs, Ys: STD_LOGIC_VECTOR (d_w_c - 1 downto 0); -- Declaring input and
output signals
Signal OpSelCs: CLB_Operation; -- Declaring the operation selection signal as an
input vector

begin
CLB_Inst: entity work.Comparison_Logic_Block(Behavioral)
    Generic map (d_w => d_w_c) -- Port-mapping the data width values of the
instatiated logic block
```

```vhdl
        Port map (A => As, -- Port-mapping the the input, output and select signals
                  B => Bs,
                  OpselC => OpselCs,
                  Y => Ys
                  );

stim: process
    begin
        As <= "011"; Bs <= "010"; -- Assigning first set of test vectors
        OpselCs <= cGTH; wait for 5 ns;
        OpselCs <= cLTH; wait for 5 ns;
        OpselCs <= cEQT; wait for 5 ns;
        OpselCs <= cGTET; wait for 5 ns;
        OpselCs <= cLTET; wait for 5 ns;
        OpselCs <= cNET; wait for 5 ns;
        OpselCs <= NoOp; wait for 5 ns;
    finish;
end process;

end Behavioral;
```