```
----------------------------------------------------------------------
--------------
-- Company:
-- Engineer:
--
-- Create Date: 02/01/2024 02:05:09 PM
-- Design Name:
-- Module Name: compunit - 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.std_logic_unsigned.ALL;
USE IEEE.numeric_std.ALL;
--use IEEE.std_logic_unsigned.all;
-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity compunit is
```

```
 Port (a,b:in std_logic_vector(7 downto 0);
  s: in std_logic_vector(3 downto 0);
  o:out std_logic_vector(7 downto 0));
end compunit;

architecture Behavioral of compunit is




begin


process (A,B,s)
begin

case s is

when"0000"=>o<=A and B;
when"0001"=>o<=A or B;
when"0010"=>o<=A nand B;
when"0011"=>o<=A nor B;
when"0100"=>o<=A xor B;
when"0101"=>o<=A xnor B;
when"0110"=>o<=A + B;
when"0111"=>o<=A - B;
when"1000"=>o<=
std_logic_vector(to_unsigned(to_integer(unsigned(a)) *
to_integer(unsigned(b)),8));
when"1001"=> if(A > B) then  o <= "11111111"; else o <=
"00000000";end if;
when"1010"=> if(A < B) then o <= "11111111"; else o <=
"00000000";end if;
when"1011"=> if(A = B) then o <= "11111111"; else o <=
"00000000";end if;
when"1100"=> if(A >= B) then o <= "11111111"; else o <=
"00000000";end if;
```

```
when"1101"=> if(A <= B) then o <= "11111111"; else o <=
"00000000";end if;
when"1110"=> if(A /= B) then o <= "11111111"; else o <=
"00000000";end if;
when"1111"=> o <="00000000";
when others => o <= (others => '0');
end case;

end process;
end Behavioral;
```