

Design:

```
library ieee;

use ieee.std_logic_1164.all;

entity eightbitcomperator is
port( A,B: in std_logic_vector(7 downto 0);
C: in std_logic_vector(3 downto 0);
RESULT:out std_logic_vector(7 downto 0));
end eightbitcomperator;

architecture eightcom_arch of eightbitcomperator is
signal COM_MIN: std_logic_vector (7 downto 0);
signal COM_MAX: std_logic_vector (7 downto 0);
begin
process(A,B)
begin
if ((A(7)='1') and (B(7)='0')) then
COM_MAX<=B;
COM_MIN<=A;
elsif ((A(7)='0') and (B(7)='1')) then
COM_MAX<=A;
COM_MIN<=B;
elsif (A(7)=B(7)) then
for i in 6 downto 0 loop
if (A(i)='0') then
if ((A(i)='0') and (B(i)='1')) then
COM_MAX<=B;
COM_MIN<=A;
exit;
elsif ((A(i)='1') and (B(i)='0')) then
COM_MAX<=A;
```

```

    COM_MIN<=B;

    exit;

    end if;

elseif(A(7)='1') then
    if ((A(i)='0') and (B(i)='1')) then
        COM_MAX<=B;

        COM_MIN<=A;

        exit;

        elsif((A(i)='1') and (B(i)='0')) then
            COM_MAX<=A;

            COM_MIN<=B;

            exit;

        end if;

    end if;

    end loop;

    end if;

end process;

with C select RESULT <=
    COM_MIN when "0110",
    COM_MAX when "0111",
    "00000000" when others;

end eightcom_arch;

```

test bench:

```

library ieee;

use ieee.std_logic_1164.all;

entity eightbitcompaeratorTB is

end eightbitcompaeratorTB;

architecture eightcomTB_arch of eightbitcompaeratorTB is

    component eightbitcomperator is

```

```

port( A,B: in std_logic_vector(7 downto 0);
C: in std_logic_vector(3 downto 0);
RESULT:out std_logic_vector(7 downto 0));
end component;

signal A : std_logic_vector(7 downto 0) := "00000000";
signal B : std_logic_vector(7 downto 0) := "00000000";
signal C: std_logic_vector( 3 downto 0) := "0000";
signal RESULT: std_logic_vector(7 downto 0);

begin

U1: eightbitcomperator port map(A,B,C,RESULT);

process

begin

C<="0110";

A<= "00000000";B <= "11111110"; wait for 10 ns;
A <= "00110010";B<= "00110010"; wait for 10 ns;
A <= "00110010";B <= "01001011";wait for 10 ns;
A <= "11001110";B <= "10110101";wait for 10 ns;
A <= "01100100";B <= "10011100";wait for 10 ns;

C<="0111";

A<= "00000000";B <= "11111110"; wait for 10 ns;
A <= "00110010";B<= "00110010"; wait for 10 ns;
A <= "00110010";B <= "01001011";wait for 10 ns;
A <= "11001110";B <= "10110101";wait for 10 ns;
A <= "01100100";B <= "10011100";wait for 10 ns;

wait;

end process;

end eightcomTB_arch;

```