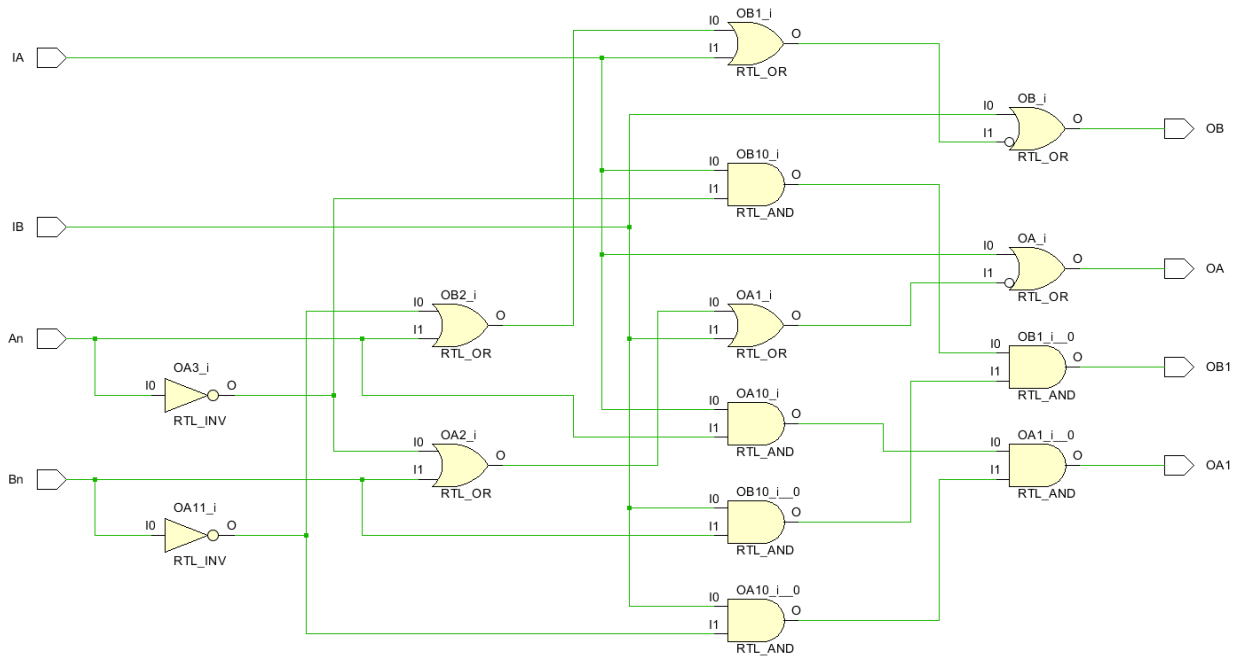
Comparator

Comparator Latch – Design Source File

```
----------------------------------------------------------------------------------

-- Group 17
-- Senevirathne S.M.P.U.
--
-- Create Date: 04/13/2024 10:02:06 PM
-- Design Name: Comparator
-- Module Name: Comparator_Latch - Behavioral
-- Project Name: Nanoprocessor
-- Target Devices: Basys3 Board

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


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity Comparator_Latch is
    Port ( An : in STD_LOGIC; --n th bit of number A
           Bn : in STD_LOGIC; --n th bit of number B
           IA : in STD_LOGIC; --input status: A
           IB : in STD_LOGIC; --input status: B
```

```
        OA : out STD_LOGIC; --output status: A
        OA1 : out STD_LOGIC; --AGreat Special Case
        OB : out STD_LOGIC; --output status: B
        OB1 : out STD_LOGIC); --BGreat Special Case
end Comparator_Latch;

architecture Behavioral of Comparator_Latch is

begin
    OA <= IA OR NOT(NOT(An) OR Bn OR IB); -- output status: A
    OB <= IB OR NOT(NOT(Bn) OR An OR IA); -- output status: B

    OA1 <= (IA AND An) AND (IB AND NOT(Bn)); --AGreat Special Case
    OB1 <= (IA AND NOT(An)) AND (IB AND Bn); --BGreat Special Case
end Behavioral;
```

Comparator Latch – Schematic Diagram



Comparator – Design Source File

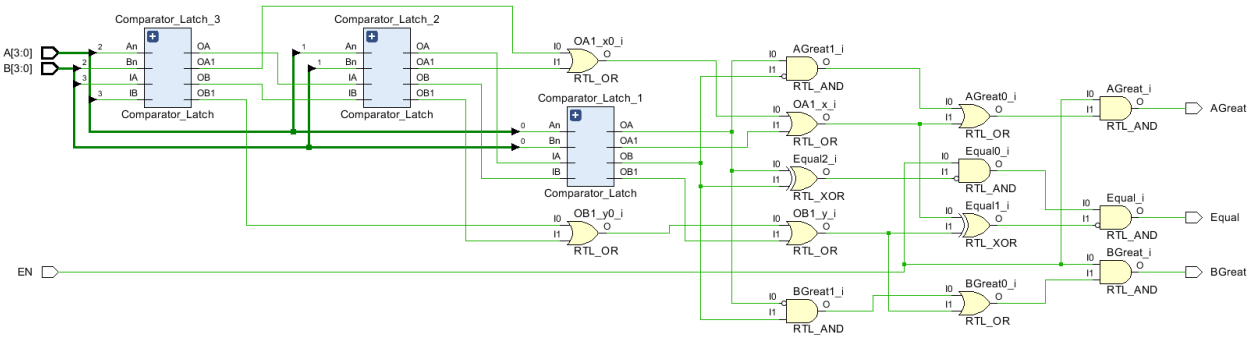
```
----------------------------------------------------------------------------------

-- Group 17
-- Senevirathne S.M.P.U.
--
-- Create Date: 04/13/2024 10:02:06 PM
-- Design Name: Comparator
-- Module Name: Comparator - Behavioral
-- Project Name: Nanoprocessor
-- Target Devices: Basys3 Board

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


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity Comparator is
    Port ( EN : in STD_LOGIC; --Enable input
           A : in STD_LOGIC_VECTOR (3 downto 0); --4 bit value: A
           B : in STD_LOGIC_VECTOR (3 downto 0); --4 bit value: B
           AGreat : out STD_LOGIC; --Is A the greater number?
           BGreat : out STD_LOGIC; --Is B the greater number?
           Equal : out STD_LOGIC); --Are the numbers equal?
end Comparator;
```

```
architecture Behavioral of Comparator is
component Comparator_Latch
   Port ( An : in STD_LOGIC; --n th bit of number A
      Bn : in STD_LOGIC; --n th bit of number B
      IA : in STD_LOGIC; --input status: A
      IB : in STD_LOGIC; --input status: B
      OA : out STD_LOGIC; --output status: A
      OA1 : out STD_LOGIC; --AGreat Special Case
      OB : out STD_LOGIC; --output status: B
      OB1 : out STD_LOGIC); --BGreat Special Case
end component;

SIGNAL IA3, IA2, IA1, IB3, IB2, IB1 : std_logic; --signal input statuses of A and B
SIGNAL OA3, OA2, OA1, OB3, OB2, OB1, OA1_3, OA1_2, OA1_1, OB1_3, OB1_2, OB1_1 :
std_logic; --signal output statuses of A and B
SIGNAL An2, An1, An0, Bn2, Bn1, Bn0 : std_logic; --signal 2 downto 0 bits of A and B
SIGNAL OA1_x, OB1_y :std_logic; --special outputs
begin
   Comparator_Latch_3 : Comparator_Latch --mapping the bottom latch
   port map (
      An => An2,
      Bn => Bn2,
      IA => IA3,
      IB => IB3,
      OA => OA3,
      OA1 => OA1_3,
      OB => OB3,
      OB1 => OB1_3
   );
   Comparator_Latch_2 : Comparator_Latch --mapping the middle latch
```

```
    port map (
        An => An1,
        Bn => Bn1,
        IA => IA2,
        IB => IB2,
        OA => OA2,
        OA1 => OA1_2,
        OB => OB2,
        OB1 => OB1_2
    );
    Comparator_Latch_1 : Comparator_Latch --mapping the top latch
    port map (
        An => An0,
        Bn => Bn0,
        IA => IA1,
        IB => IB1,
        OA => OA1,
        OA1 => OA1_1,
        OB => OB1,
        OB1 => OB1_1
    );
--Assigning the inputs to the bottom latch
IA3 <= B(3);
IB3 <= A(3);
An2 <= A(2);
Bn2 <= B(2);

--Assigning the inputs to the middle latch
IA2 <= OA3;
```

```
IB2 <= OB3;

An1 <= A(1);

Bn1 <= B(1);


--Assigning the inputs to the top latch

IA1 <= OA2;

IB1 <= OB2;

An0 <= A(0);

Bn0 <= B(0);


--Defining OA1_x and OA1_y for the special cases

OA1_x <= OA1_3 OR OA1_2 OR OA1_1;

OB1_y <= OB1_3 OR OB1_2 OR OB1_1;


--Defining the flags; AGreat, BGreat and Equal

AGreat <= EN AND ((OA1 AND NOT(OB1)) OR OA1_x);

BGreat <= EN AND ((NOT(OA1) AND OB1) OR OB1_y);

Equal <= EN AND NOT(OA1 XOR OB1) AND NOT(OA1_x XOR OB1_y);


end Behavioral;
```

Comparator – Schematic Diagram



Comparator – Simulation File

```
----------------------------------------------------------------------------------

-- Group 17
-- Senevirathne S.M.P.U.
--
-- Create Date: 04/13/2024 10:02:06 PM
-- Design Name: Comparator
-- Module Name: Comparator_TB - Behavioral
-- Project Name: Nanoprocessor
-- Target Devices: Basys3 Board

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

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity Comparator_TB is
end Comparator_TB;

architecture Behavioral of Comparator_TB is
    component Comparator is
        Port ( EN : in STD_LOGIC; --Enable input
               A : in STD_LOGIC_VECTOR (3 downto 0); --4 bit value: A
               B : in STD_LOGIC_VECTOR (3 downto 0); --4 bit value: B
               AGreat : out STD_LOGIC; --Is A the greater number?
               BGreat : out STD_LOGIC; --Is B the greater number?
```

```
        Equal : out STD_LOGIC); --Are the numbers equal?
    end component;

    --Signal EN, A, B, AGreat, BGreat and Equal
    SIGNAL EN : std_logic;
    SIGNAL A : std_logic_vector (3 downto 0);
    SIGNAL B : std_logic_vector (3 downto 0);
    SIGNAL AGreat : std_logic;
    SIGNAL BGreat : std_logic;
    SIGNAL Equal : std_logic;



begin
    Comparator_Sim : Comparator --mapping comparator for simulation
        port map(
            EN => EN,
            A => A,
            B => B,
            AGreat => AGreat,
            BGreat => BGreat,
            Equal => Equal);

    process begin
        wait for 100ns;
        EN <= '1';

        A <= "0000"; -- A = 0

        B <= "0000"; -- B = 0, Equal
```

```
wait for 100ns;

B <= "0111"; -- B = 7, BGreater
wait for 100ns;

B <= "1000"; -- B = -8, AGreater
wait for 100ns;

EN <= '0';

A <= "0111"; -- A =7

B <= "0000"; -- B = 0, AGreater
wait for 100ns;

B <= "0111"; -- B = 7, Equal
wait for 100ns;

B <= "1000"; -- B = -8, AGreater
wait for 100ns;

EN <= '1';

A <= "1000"; -- A = -8

B <= "0000"; -- B = 0, BGreater
wait for 100ns;

B <= "0111"; -- B = 7, BGreater
```

```
wait for 100ns;

B <= "1000"; -- B = -8, Equal
wait for 100ns;

EN <= '0';

A <= "0000"; -- A = 0

B <= "0000"; -- B = 0, Equal
wait for 100ns;

B <= "0111"; -- B = 7, BGreater
wait for 100ns;

B <= "1000"; -- B = -8, AGreater
wait for 100ns;

EN <= '1';

A <= "0111"; -- A =7

B <= "0000"; -- B = 0, AGreater
wait for 100ns;

B <= "0111"; -- B = 7, Equal
wait for 100ns;

B <= "1000"; -- B = -8, AGreater
```

```
        wait for 100ns;

        EN <= '0';

        A <= "1000"; -- A = -8

        B <= "0000"; -- B = 0, BGreater
        wait for 100ns;

        B <= "0111"; -- B = 7, BGreater
        wait for 100ns;

        B <= "1000"; -- B = -8, Equal
        wait for 100ns;

        EN <= '1';
        B <= "1010"; -- B = -6, BGreater

        wait;
    end process;

end Behavioral;
```

Comparator – Timing Diagram