```vhdl
----------------------------------------------------------------------------------
---------------
-- Company: 
-- Engineer: 
-- 
-- Create Date: 17.04.2023 14:37:28
-- Design Name: 
-- Module Name: ALU_1 - 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;
use ieee.NUMERIC_STD.ALL;
use ieee.std_logic_unsigned.all;

entity ALU is

 Port (N:IN STD_LOGIC_VECTOR (3 DOWNTO 0);       --CU input1
       R:IN STD_LOGIC_VECTOR (3 DOWNTO 0);       --CU input2
       Sel:IN STD_LOGIC_VECTOR (4 DOWNTO 0);     --CU Selector
       S:OUT STD_LOGIC_VECTOR (3 DOWNTO 0);      --CU output
       COUT: OUT STD_LOGIC                       --carry out
 );
end ALU;
```

```
architecture Behavioral of ALU is
SIGNAL Q: STD_LOGIC_VECTOR (4 DOWNTO 0);

BEGIN
PROCESS(N,R,Sel)IS
BEGIN

CASE Sel IS

 WHEN "00000" =>    --for addition
 S<=N+R;

 WHEN "00001" =>    --for subtraction
 S<=N-R;

 WHEN "00010" =>    --for multiplication
 S<=std_logic_vector(to_unsigned(to_integer(unsigned(N)) *
to_integer(unsigned(R)),4));

 WHEN "00100" =>    --for greater than
 IF (N>R) THEN
 S<="1111"; ELSE
 S<="0000";
 END IF;

 WHEN "00011" =>    --for equal to
 IF (N=R) THEN
 S<="1111"; ELSE
 S<="0000";
 END IF;

 WHEN "00101" =>    --for less than
 IF(N<R) THEN
 S<="1111"; ELSE
 S<="0000";
```

```
END IF;

WHEN "00110" =>    --for OR
S<= N OR R;

WHEN "00111" =>    --for NOR
S<= N NOR R;

WHEN "01000" =>    --for NAND
S<= N NAND R;

WHEN "01001" =>    --for AND
S<= N AND R;

WHEN "01010" =>    --for XOR
S<= N XOR R;

WHEN "01011" =>    --for XNOR
S<= N XNOR R;

WHEN "01100" =>    --ROTATE RIGHT
S<=to_stdlogicvector(to_bitvector(N) ROR to_integer(unsigned(R)));

WHEN "01101" =>    --ROTATE LEFT
S<=to_stdlogicvector(to_bitvector(N) ROL to_integer(unsigned(R)));

WHEN "01110" =>    --ARITHMETIC SHIFT LEFT
S<=to_stdlogicvector(to_bitvector(N) SLA to_integer(unsigned(R)));

WHEN "01111" =>    --LOGICAL SHIFT RIGHT
S<=to_stdlogicvector(to_bitvector(N) SRL to_integer(unsigned(R)));

WHEN "10000" =>    --LOGICAL SHIFT LEFT
S<=to_stdlogicvector(to_bitvector(N) SLL to_integer(unsigned(R)));



WHEN "10001" =>
```

```
 S<= N;             -- PASS GATE 1

 WHEN "10010" =>
 S<= R;             -- PASS GATE 2

 WHEN "10011" =>    -- NOOP
 S<= "0000";

 WHEN OTHERS =>     --if any other bits are given to the selector's
other than mentioned above the output is "0000"
 S<="0000";

 END CASE;
 END PROCESS;

 Q<=STD_LOGIC_VECTOR ('0'&N)+ STD_LOGIC_VECTOR ('0'&R);
 COUT<=Q(4);

end Behavioral;
```