Practical 1 Part A ALU

Code

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
---- Uncomment the following library declaration if instantiating
---- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;
entity bham1 is
    Port ( a : in  STD_LOGIC_VECTOR (3 downto 0);
           b : in  STD_LOGIC_VECTOR (3 downto 0);
  Sel : in  STD_LOGIC_VECTOR (3 downto 0);
           Y : out  STD_LOGIC_VECTOR (3 downto 0));
end bham1;
architecture Behavioral of bham1 is
begin
process(a,b,Sel)
begin
case Sel is
when "0000"=> Y<=a+b;
when "0001"=> Y<=a-b;
when "0010"=> Y<=a and b;
when "0011"=> Y<=a nand b;
when "0100"=> Y<=a xor b;
when "0101"=> Y<=a xnor b;
when "0110"=> Y<=a or b;
when "0111"=> Y<=a;
when "1000"=> Y<=b;
when others => Y<="0000";
end case;
end process;
end Behavioral;
```

TestBench

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.std_logic_unsigned.all;
USE ieee.numeric_std.ALL;

ENTITY bhamare1_vhd IS
END bhamare1_vhd;

ARCHITECTURE behavior OF bhamare1_vhd IS

	-- Component Declaration for the Unit Under Test (UUT)
	COMPONENT bham1
	PORT(
		a : IN std_logic_vector(3 downto 0);
		b : IN std_logic_vector(3 downto 0);
		Sel : IN std_logic_vector(3 downto 0);
		Y : OUT std_logic_vector(3 downto 0)
		);
	END COMPONENT;

	--Inputs
	SIGNAL a :  std_logic_vector(3 downto 0) := (others=>'0');
	SIGNAL b :  std_logic_vector(3 downto 0) := (others=>'0');
	SIGNAL Sel :  std_logic_vector(3 downto 0) := (others=>'0');

	--Outputs
	SIGNAL Y :  std_logic_vector(3 downto 0);

BEGIN

	-- Instantiate the Unit Under Test (UUT)
	uut: bham1 PORT MAP(
		a => a,
```

```
                b => b,
                Sel => Sel,
                Y => Y
        );

        tb : PROCESS
        BEGIN

                -- Wait 100 ns for global reset to finish
                wait for 100 ns;

                a<="1011";
  b<="1001";
  Sel<="0000";--Result of Addition
  wait for 100 ns;
  Sel<="0001";--Result of Subtraction
  wait for 100 ns;
  Sel<="0010";--Result of and
  wait for 100 ns;
  Sel<="0011";--Result of xor
  wait for 100 ns;
  Sel<="0100";--Result of xnor
  wait for 100 ns;
  Sel<="0101";--Result of or
  wait for 100 ns;
  Sel<="0111";--Value of a
  wait for 100 ns;
  Sel<="1000";--Value of b
  wait for 100 ns;
  wait for 100 ns;
 wait; -- will wait forever
 END PROCESS;

 END;
```