**Lab Objective:** Introducing unsigned multipliers, using add and shift operations. 

**Multiplication of Unsigned Integer Numbers, using add and shift**

Multiplication involves the generation or partial products, one for each digit in the multiplier. These partial products are then summed to produce the final product. The multiplication of two n-bit binary integers results in a product of up to 2n bits in length (e.g., 11 x 11 = 1001)

### Algorithm to implement the multiplication operation:

1. Assume a double width register. Put the multiplier in the least significant half, and clear the most significant half.
2. If there are n bits in a single width register, perform the following n times:
   a. If the least significant bit of the register contains a 1, add the multiplicand to the most significant half.
   b. Shift the whole register one bit to the right, throwing away the least significant bit and shifting the carry bit into the least significant bit.
**Example:-** The VHDL code for a 4-Bit X 4-Bit unsigned multiplier, which is based on the above algorithm.

**Important Notes:-**
1. To perform the addition use + sign.
2. To perform the shifting use the concatenating sign &. Example A(0 to 3) ="0001″ → '0'&A(0 to 3) ="0000″. The last element A(3) is gone.
3. The variable in the architecture part is different from the signal in the entity part. The signal can have delay, while the variable can not have delay.
4. The IEEE Packages are pre-compiled VHDL code.

**Num1 → Multiplicand → 101**
**Num2 → Multiplier → 111**
**Product → 100011**

**Solutions:-**

```vhdl
Library IEEE;
Use IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;

Entity multiplication_e Is
Port(
  Num1,Num2:in std_logic_vector(2 downto 0);
  product:out std_logic_vector(5 downto 0);
);
End Entity;

Architecture multiplication_a of multiplication_e is
Begin
  Process(Num1,Num2)
  variable reg:std_logic_vector(5 downto 0);
  variable add:std_logic_vector(3 downto 0);
  Begin
    reg:="000"&Num2;
    For i in 1 to 3
      Loop
        If reg(0)='1' then
          add:=('0'&num1)+('0'&reg(5 downto 3));
          reg:=add&reg(2 downto 1);
        Else
          reg:='0'&reg(5 downto 1);
        End If;
      End Loop;
      product<=reg;
  End Process;
End Architecture;
```
**Assignmet-2:** Write the VHDL code for a 32-Bit X 32-Bit Unsigned Multiplier, and then simulate it. Don’t use descending order for vectors, use ascending order.

### Important notes:-

1. Two students can work on the assignment.
2. The submission date is two weeks from now.
3. Don’t submit your work by email.
4. Submit only a hard copy.
5. Don’t copy other student’s work.