**Briefly explain**

 In this experiment we a counter using Quartus on known frequency and display the digit on seven segment display . To do that we divided our work as **Frequency –divider** , **4-bit counter** **and Seven segment** after that we connected them together.

**\_Frequency-divider:**

 Because we face high frequency (27,50)MHz ,the simulation was useless . So we built the frequency-divider that change from high to low frequency .In this experiment we reduced the frequency from 27 MHz to 2 Hz.

**\_4-bit counter:**

 We built our 4-bit counter which work on positive edge and reset on negative edge.

**Unsigned multiplication :**\_

 One of the ways to multiply two positive numbers is using unsigned multiplication that receives two input and multiply them to give the output after. Its algorithm based on shift and add.

**Booth multiplication:**\_

 In booth multiplication we could multiply any two positive or negative numbers . Its algorithm depend on shift , add and subtract .

**Procedure :**

**\_4-bit counter:**

We wrote the code shown in fig (1) after that we simulate as in fig (2) .finally we created the symbol.



Fig (1):code for 4-bit counter



Fig(2):simulation for fig 1

**Frequency-divider:** \_

we could change the speed of the counter by changing the frequency rate . for example if we want it to be slow we increase the ratio. the code for our freq\_divider shown in fig (3).



Fig(3):code for frequency divider

\_**Seven segment:**

The seven segment on FPGA was active low so we replaced the 1's by 0's and 0's by 1's in the code of lab manual as shown in fig (4).



Fig(4):code for seven segment

After that we connected them together as shown in fig (5) and put the final result on FPGA .



Fig(5):couter system

\_ **Unsigned multiplier:**

We wrote the code shown in fig(6), 

Fig(6):code for unsigned multiplication



Fig(7):simulation for unsigned multiplication

**Booth multiplication:**

The code of the booth is like unsigned multiplication but with small diffrance as in fig(8).fig(8):code for booth algorithm



Fig(9):simulation for booth

**Conclusion:**

At the end of this experiment, we have used the counter as an input to the design that we have, we also have learned how we can adjust the frequency as we desire it to be using the div\_freq code. We have also implemented some algorithms using vhdl language, in this case the unsigned multiplication and booth algorithm, where we have taken an idea about how we can implement different algorithms using VHDL language, which makes it easy to compile and simulate using quartos, so after that the synthesis of this code will be achieved easily.