Lab Report

Lab Session-8

MATHEW M PHILIP EE22BTECH11211

JAY VIKRANT EE22BTECH11025

AIM: Build an Asynchronous counter and verify the functionality. You may have to use Arduino to get low frequency clock to verify.

COMPONENT: LED(x4), Wires, Arduino UNO, IC 7476

THEORY: The counter is an electronic device that can count from 0 to 15. We use an asynchronous counter to build this. An asynchronous counter also known as a ripple counter is a type of counter that can count in a binary sequence using a series of flip-flops. Unlike a synchronous counter, where a common clock signal is fed to all the flip-flops, an asynchronous counter relies on the output of the previous flip-flop to act as the clock to the next flip-flop.

The basic building block of an asynchronous counter is JK flip-flop: it has two inputs (J input and a K input) and a clock input. The clock input controls when the flip flop updates its outputs based on the input data. The output of one flip-flop is connected to the clock input of the next flip-flop in sequence.

To create an n-bit counter, n flip-flops are to be connected in series. The output of the first flip-flop represents the Least Significant Bit (LSB) and the output of the last flip-flop represents the Most Significant Bit (MSB). This causes a ripple effect where each flip-flop changes its state one after the other, resulting in a binary counting sequence.

The JK flip-flops must be used in a toggle state to get the required output. So, we connect both J and K inputs to the “1” or the high state. So, in each negative edge pulse transition the output of the D flip-flop is the complement of the previous output.

$$Q(n+1) = Q'(n)$$

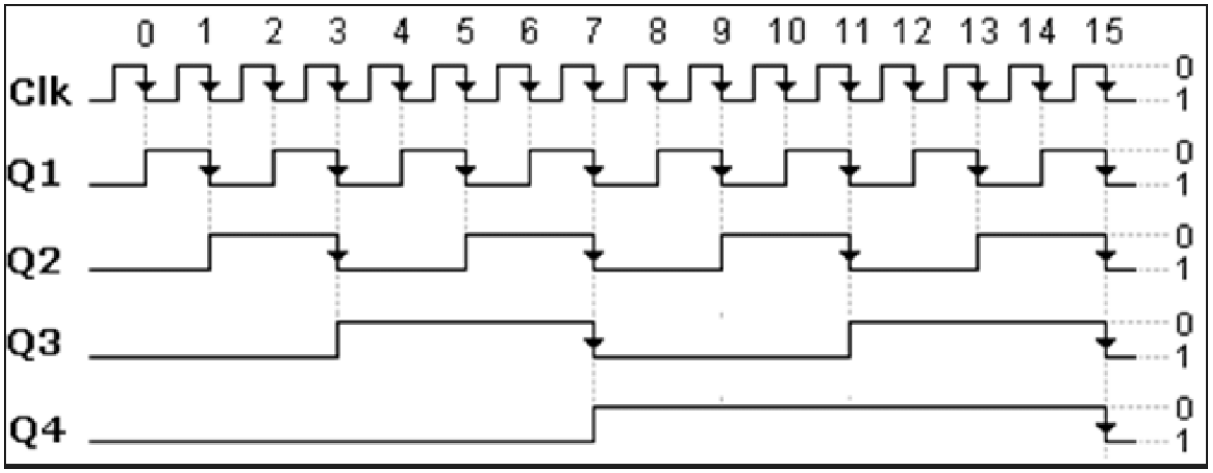
The outputs of each of the flip-flop is clock wave having frequency half of that of the clock input of the respective flip-flop. i.e., If the frequency of the clock=f.

The frequency of the first output wave=f/2

The frequency of second output wave=f/4

The frequency of third output wave=f/8

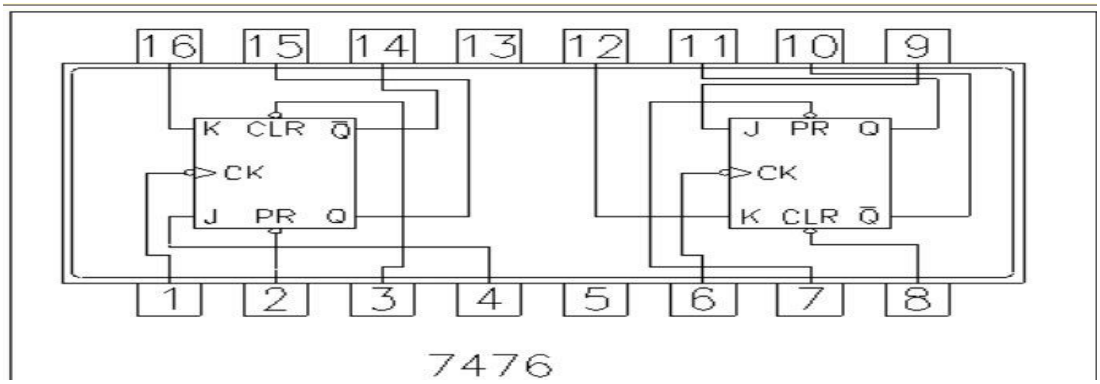
The frequency of fourth output wave=f/16



The above is the timing diagram for the asynchronous ripple counter. We can see that for each output the frequency gets halved or the time-period doubles.

Table 1: Truth table for a 4-bit synchronous counter using JK flip-flops

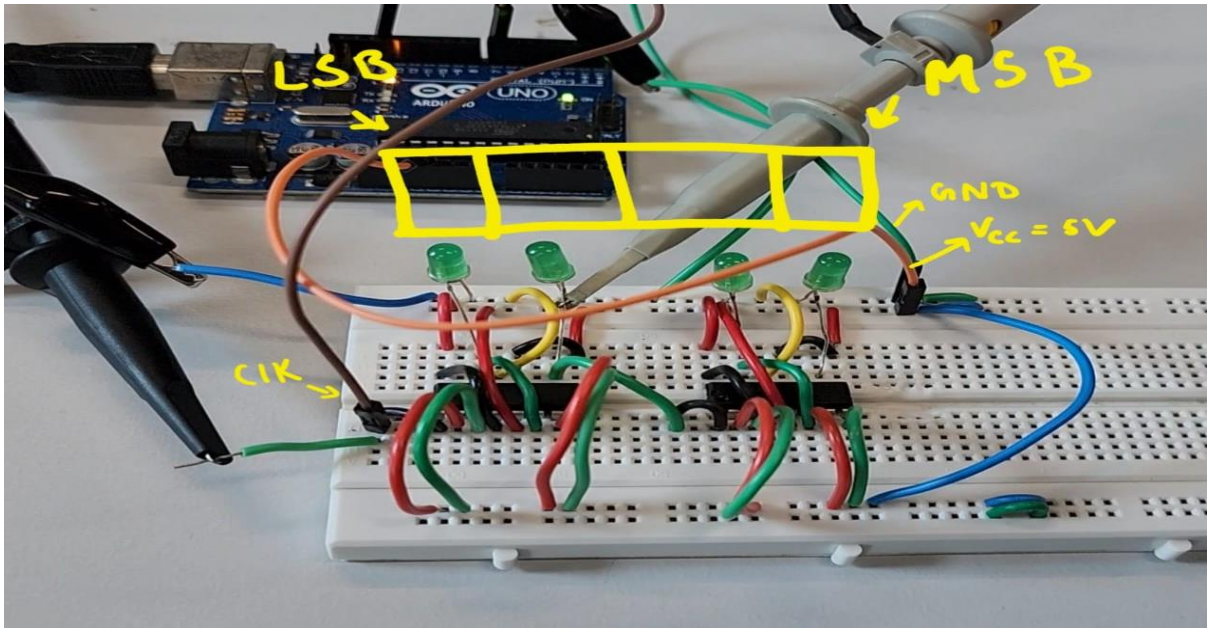
Clock	Q0	Q1	Q2	Q3	Next State
0	0	0	0	0	0000
1	0	0	0	1	0001
1	0	0	1	0	0010
1	0	0	1	1	0011
1	0	1	0	0	0100
1	0	1	0	1	0101
1	0	1	1	0	0110
1	0	1	1	1	0111
1	1	0	0	0	1000
1	1	0	0	1	1001
1	1	0	1	0	1010
1	1	0	1	1	1011
1	1	1	0	0	1100
1	1	1	0	1	1101
1	1	1	1	0	1110
1	1	1	1	1	1111



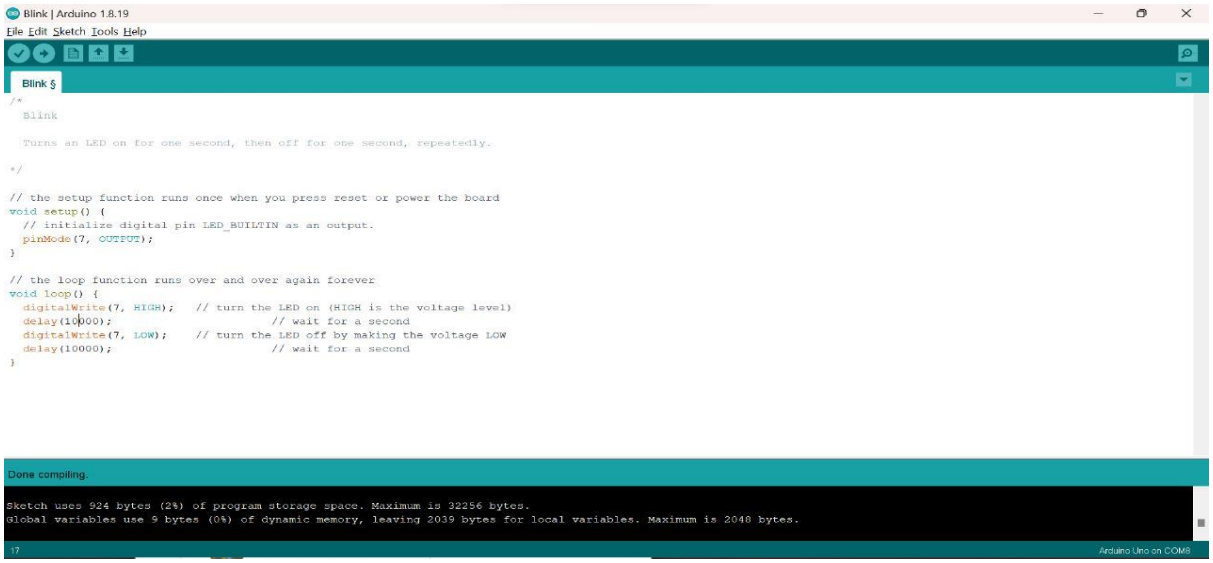
PROCEDURE: Firstly, we write the code in Arduino IDE for input clock. The clock is made using example code from Files>example>Basics>Blink in the IDE. In the code High voltage is set for pin(7) on Arduino uno, then the delay of High voltage and LOW voltage is set to 10000 units as each unit is a millisecond, for a time period of 10 seconds. Thus, resulting in 0.1 Hz square waves.

Now, the following wiring is done to achieve the desired output.

1. Connect the clock (i.e pin (7)) output of the Arduino to the clock input of first JK flip-flops.
2. Connect the reset and clear input of all four JK flip-flops to V_{cc}.
3. Connect the J and K inputs of the first JK flip-flop to V_{cc}.
4. Connect the Q' output of the first JK flip-flop to the CLK input of the second JK flip-flop of the same IC7476.
5. Connect the Q' output of the second JK flip-flop of the first IC7476 to the clock of the first JK flip-flop of the second IC7476.
6. Connect the Q' outputs of all four JK flip-flops to the LEDs.



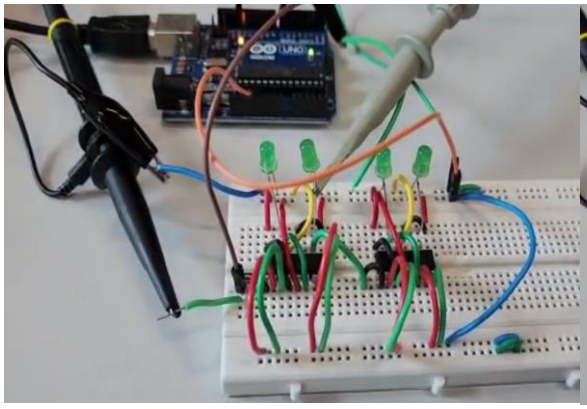
The clock has a time period of 20 seconds.



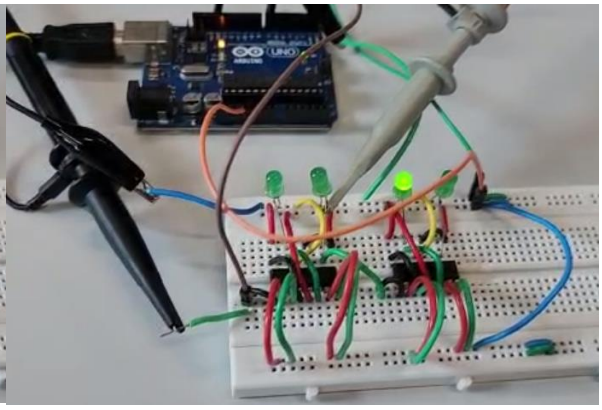
RESULT:

The following result was observed from t=0 to t=300 seconds,

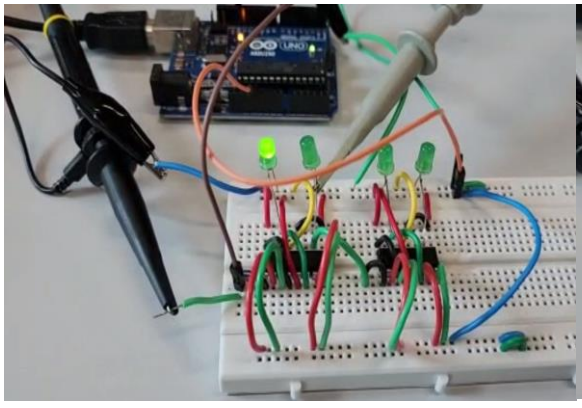
0000



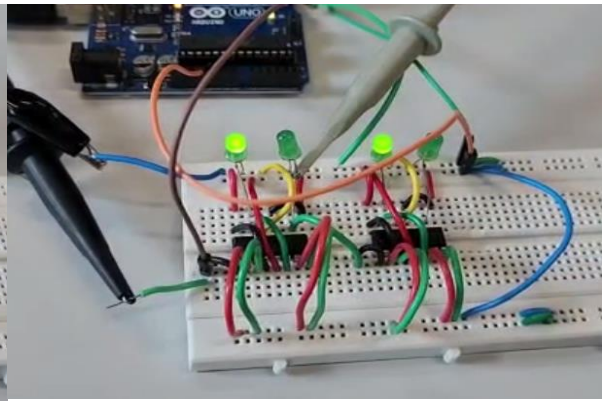
0010



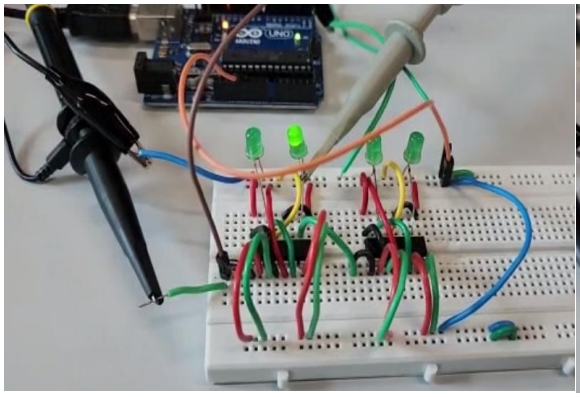
1000



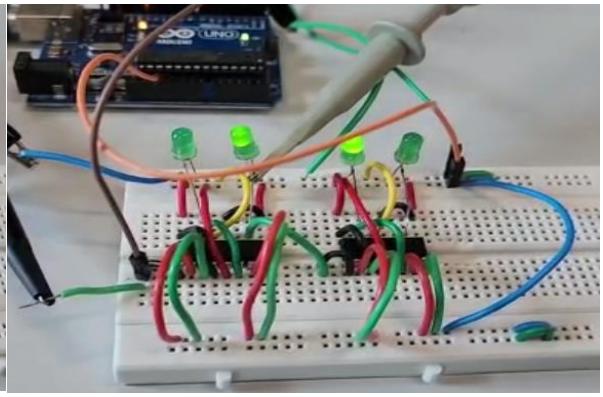
1010



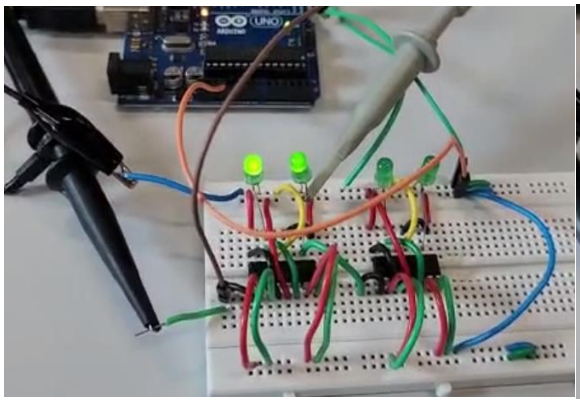
0100



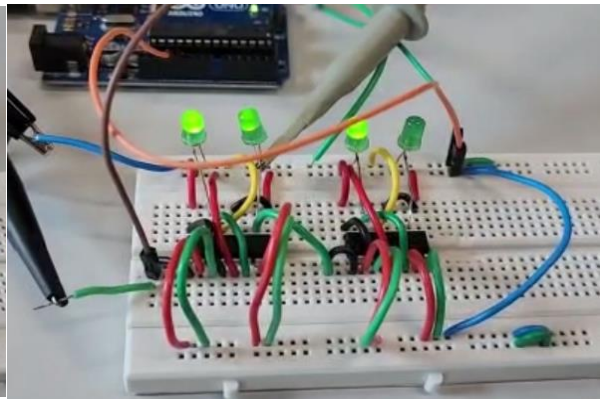
0110



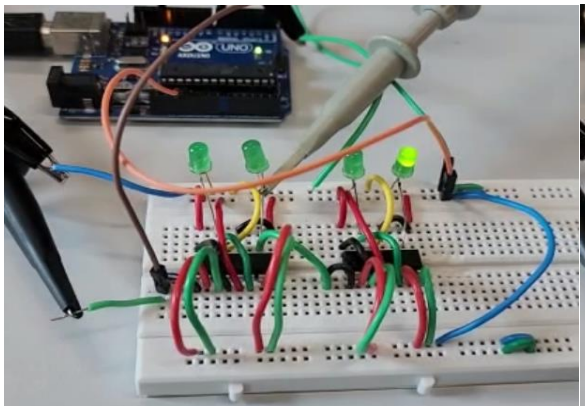
1100



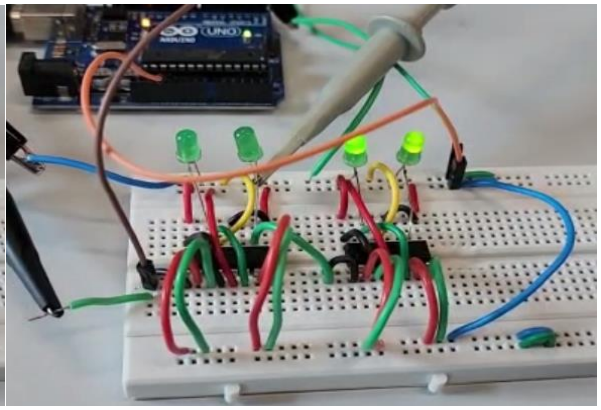
1110



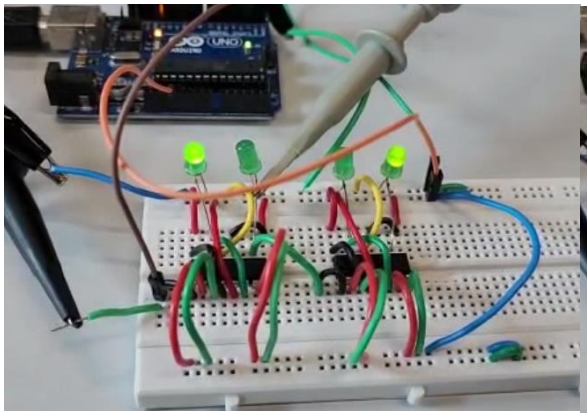
0001



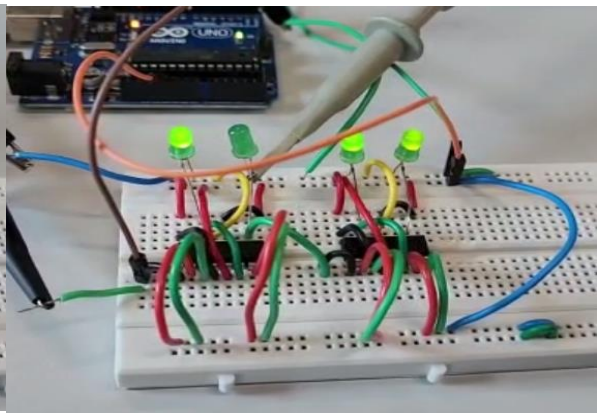
0011



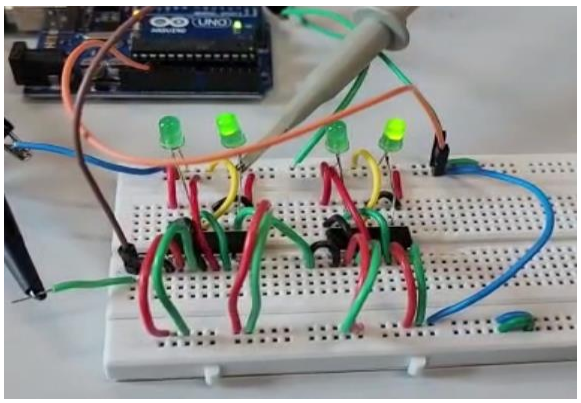
1001



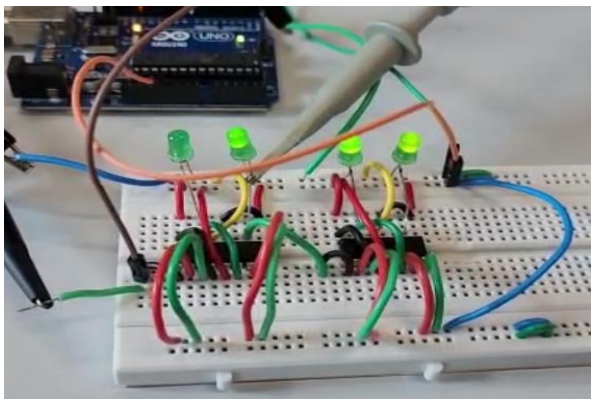
1011



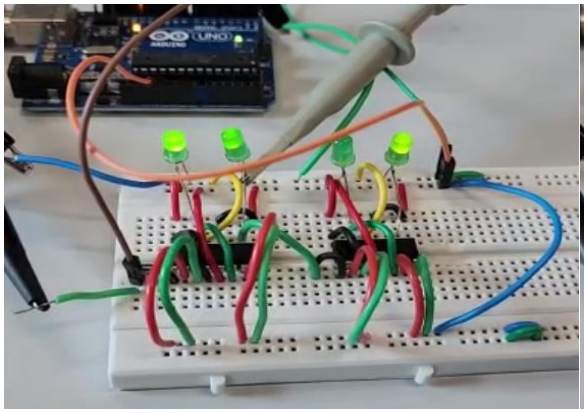
0101



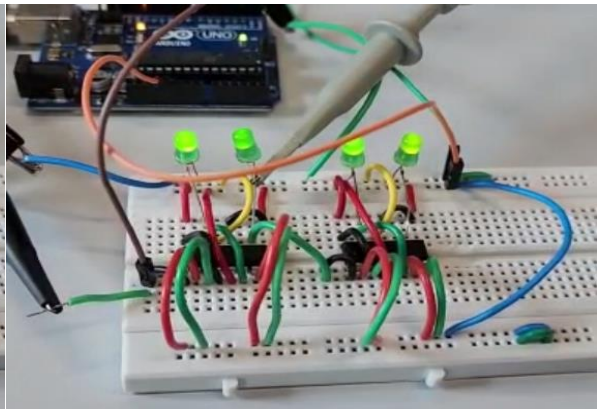
0111



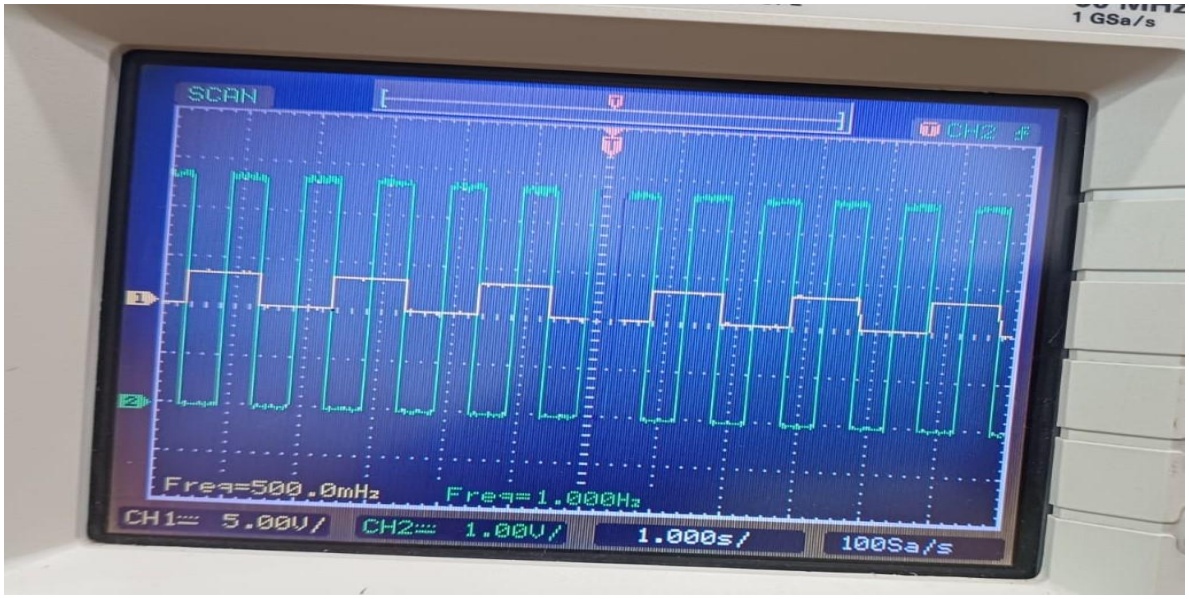
1101

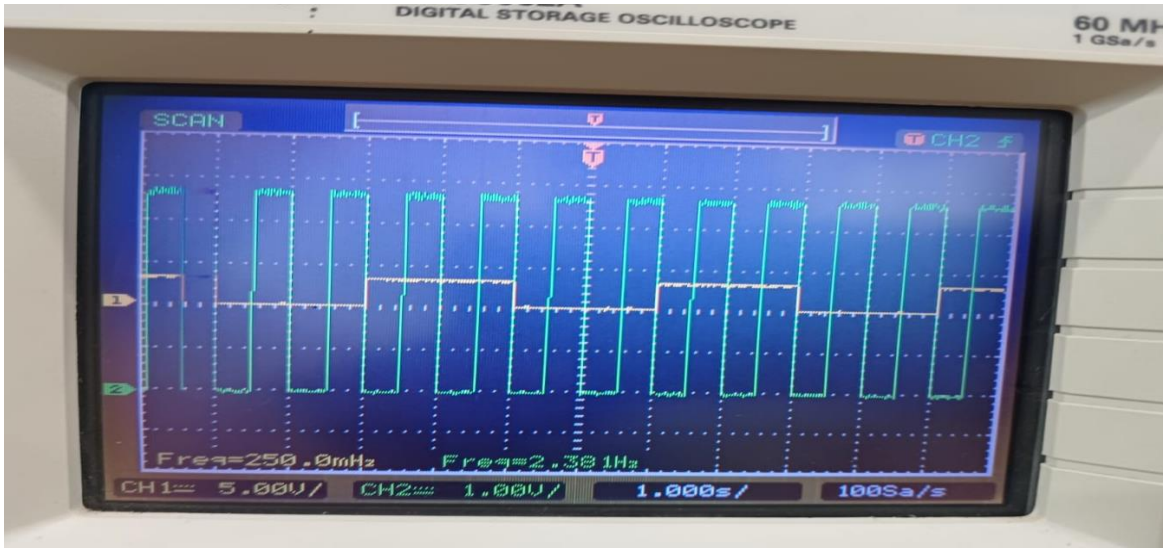


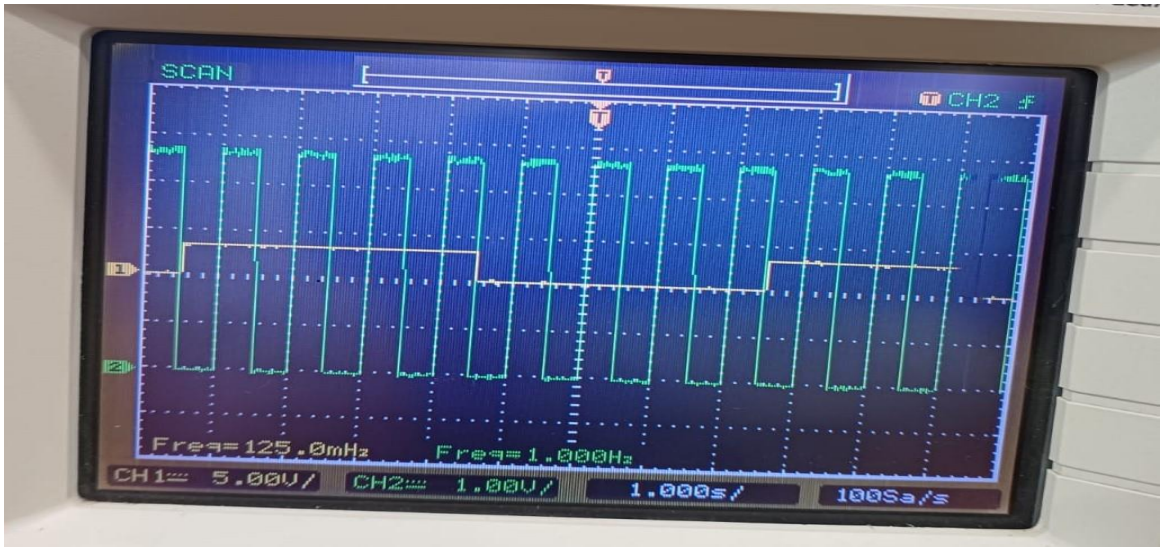
1111

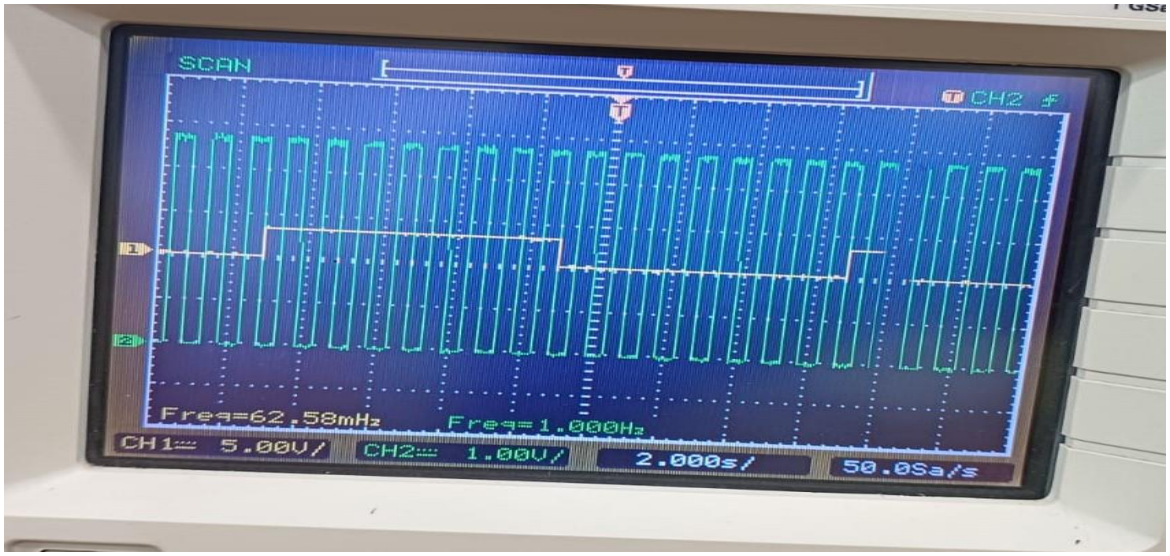


Also, the following result was seen in the oscilloscope for each output (i.e the LED from left to right)


1 GSa/s
SCAN
CH2
Freq=500.0mHz
Freq=1.000Hz
CH1 5.00V/
CH2 1.00V/
1.000s/
100Sa/s


DIGITAL STORAGE OSCILLOSCOPE
1 GSa/s
SCAN
CH2
Freq=250.0mHz
CH1 5.00V/
CH2 1.00V/
1.000s/
100Sa/s


SCAN
CH2
Freq=125.0mHz
Freq=1.000Hz
CH1 5.00V/
CH2 1.00V/
1.000s/
100Sa/s



We observe that frequency is halved for every output from left to right.