Experiment # 8
Clock generator circuits and Counters

1. Objectives:
   1. Understanding the principles and construction of Clock generator.
   2. To be familiar with clock pulse generation using 555 timer.
   3. Introduction to counters. Design and applications.

2. Theory:

   Clock Generator:

   Timing or synchronization is very crucial to most electronic devices and systems. This is because timing is essential in maintaining the proper sequencing of events. There are many ICs designed and manufactured specifically to accomplish this task. One of the most popular of these ICs is the 555 Timer. Figure 9.1 shows the pin diagram of the 555 Timer IC.

   The 555 timer can be used in a variety of forms. The monostable mode will create a single pulse of a specified width. A stable mode will create a repeating pulse train of specified frequency and duty cycle.

   Figure 9.2 shows the 555 IC Timer configured to output a clock signal.
The charging time (output high) is given by:

\[ t_{\text{high}} = 0.693 \times (R_1 + R_2) \times C \]

And the discharging time (output low) by:

\[ t_{\text{low}} = 0.693 \times (R_2) \times C \]

Thus the total period is:

\[ T = t_{\text{high}} + t_{\text{low}} = 0.693 \times (R_1 + 2 \times R_2) \times C \]

The clock frequency is determined by the formula:

\[ F = \frac{1}{T} = \frac{1.44}{((R_1 + 2 \times R_2) \times C)} \]

Timer 555 is used to decrease debounds and control the duty of cycle.

The duty cycle (ratio of low time to entire period) may be determined from the following equation:

\[
D_0 = \frac{t_{\text{low}}}{(t_{\text{low}} + t_{\text{high}})} = \frac{R_2}{(R_1 + 2 \times R_2)}
\]

\[
D_1 = \frac{t_{\text{high}}}{(t_{\text{low}} + t_{\text{high}})} = \frac{(R_1 + R_2)}{(R_1 + 2 \times R_2)}
\]

**Counters:**

A counter is a sequential circuit that goes through a prescribed sequence of states upon the application of input pulses. The input pulses (count pulses) of the counter is clock pulses, or from some external source, and may occur at prescribed intervals of time or at random.

Counters can be classified into two broad categories according to the way they are clocked:

1. **Asynchronous (ripple) counters** – the first FF is clocked by the external clock pulse, and then each successive FF is clocked by the Q or Q’ output of the previous FF.
2. **Synchronous counters** – all FFs are simultaneously triggered by the same clock.

The counter follows the binary number sequence or other sequence of states. A counter that follows the binary sequence is called a binary counter. An n-bit binary counter consists of n flip-flops and can count in binary from 0 to \(2^n-1\).
First: Synchronous counters

I- Johnson counter
Synchronous counter where the complement of the output of the last shift register is connected to the input of the first register and circulates a stream of ones followed by zeros around the ring as shown in figure 9.3

II- Ring counter
A ring counter is a type of counter composed of a circular shift register. The output of the last shift register is fed to the input of the first register.
Second : Asynchronous (ripple) counters

1- **Binary Ripple Counter**

A binary ripple counter consists of a series connection of complementing flip flops, with the output of each flip flop connected to the Clk input of the next higher-order flip-flop. The flip flop holding the least significant bit receives the incoming count pulses.

- **2-bit ripple up counter**
  Q’ from one flip-flop is connected to clock input for the next flip-flop MSB.

- **3-bit ripple up counter**
- 4-bit ripple up counter (negative edge triggered)
3-bit ripple down counter

II-BCD Ripple Counter

A decimal counter follows a sequence of ten states and returns to 0 after the count of 9. Such counter must have at least four flip-flops to represent each decimal digit, since a decimal digit is represented by a binary code with at least four bits. The sequence of states in a decimal counter is indicated by the binary code used to represent a decimal digit. If BCD is used, the sequence of states is as shown in the state diagram of figure 9.4. This is similar to a binary counter, except that the state after 1001 (code for decimal 9) is 0000 (code for decimal 0).

The logic diagram of a BCD ripple counter is shown in figure 9.5. The four outputs are designed by the letter symbol Q with a numeric subscript equal to the binary weight of the corresponding bit in the BCD code. The flip-flops trigger on the negative edge. Note that the output $Q_1$ is applied to the Clk inputs of both $Q_2$ and $Q_3$ and the output of $Q_2$ is applied to the Clk input of $Q_4$. The J and K inputs are connected either to a permanent 1 signal or to outputs of flip-flops, as shown in the diagram.
The following are the conditions for each flip-flop state transition:
1. \( Q_1 \) is complemented on the negative edge of every count pulse.
2. \( Q_2 \) is complemented if \( Q_8 = 0 \) and \( Q_1 \) goes from 1 to 0.
   \( Q_2 \) is cleared if \( Q_8 = 1 \) and \( Q_1 \) goes from 1 to 0.
3. \( Q_4 \) is complemented when \( Q_2 \) goes from 1 to 0.
4. \( Q_8 \) is complemented when \( Q_4 Q_2 = 11 \) and \( Q_1 \) goes from 1 to 0.
   \( Q_8 \) is cleared if either \( Q_4 \) or \( Q_2 \) is 0 and \( Q_1 \) goes from 1 to 0.

Other presentation of BCD ripple counter will be as follows:
1. We need four flip-flops, for example JK flip flops where J and K are high.
2. Count from 0-9, so when counter reaches 9 it will make the 10 to be 0.
3. Decimal 10 = (1010)_2, clear = \( D C' B A' \)
4. Take ones as inputs of NAND and take output of NAND to CLR.

Figure 9.5: BCD ripple counter
Lab Work:

Equipments Required:

1. 74LS93 4-bit binary counter.
2. 74LS160 synchronous BCD counter.
3. 74LS190 up/down synchronous BCD counter.

Procedures:

Part I: 4-bit binary Asynchronous counter

The 7493 is a 4 stage asynchronous counter containing high speed FF. Features a master reset (MR₁ & MR₂) to override the clock and force all outputs low. The 74LS93 can be used as:

1. A 4-bit ripple counter - the output Q₀ must be externally connected to input CP₁. The input count pulses are applied to input CP₀, outputs are taken at Q₀ to Q₃.
2. A 3-bit ripple counter - the input count pulses are applied to input CP₁. The outputs are available at Q₁ to Q₃.

a) Derive the functional table of the 74x93, and verify it experimentally.
b) Set the clock to its lowest frequency of operation and connect it to the clock input of the 74x93.
c) Verify that the 74x93 behaves as a divided by 16 counter (to count 0,1,2,……,15).
d) Verify that the 74x93 can also behave as a divided by 8 counter (to count 0,1,2,……,7).
e) Design the 74x93 as a divided by 4 counter (to count 0,1,2,3) without any SSI logic.
f) Design the 74x93 as a divided by 9 counter (to count 0,1,2,……,8) without any SSI logic.

![Diagram of 74LS93 circuit](image-url)

Binary 0-15
Experiment# 8

Clock generator circuits and Counters

if \( cp_0 = q_1 \) then
    clock test on
if \( cp_1 = q_1 \) then
    output on

Binary 0-7

if \( cp_0 = q_1 \) then
    clock test on
if \( cp_1 = q_1 \) then
    output on

Binary 0-3

Binary 0-8
Part II: Synchronous BCD counter

The 74x160 is a high-speed synchronous decade counter with option for applications in programmable loading / dividers array.

Install the 74x160 in the place where you had the 74x93 and connect pins 1, 3, 4, 5, 6, 7, 9 & 10 to switches and the outputs to the decoder, verify load clear & enable input condition from data sheets.

a) Derive the functional table of the 74x160, and verify it experimentally.
b) Arrange the 74x160 as a decade counter.
c) Design the 74x160 as a divided by 8 using only a single INV gate.
d) Design the 74x160 as a divided by 6 and program the counter to have the sequence 1,2,3,4,5,6,1,2,3,….. use only a single NAND gate.
Exercise

1) Using 555 timer equations, if $R_1 = R_2 = 10k\Omega$, $C = 1\mu F$ then find:
   a) $t_{\text{low}}$
   b) $t_{\text{high}}$
   c) $T$
   d) $F$

2) Draw 4-bit ring counter using D flip flops and then find its sequence like Table 1 where $Q_A=1$, $Q_B=0$, $Q_C=0$, $Q_D=0$ at Clock Pulse No = 0.

3) Draw ripple down counter from 15 to 0 using JK flip flops with negative edge triggered.

4) Draw BCD ripple counter using D flip-flops with negative edge triggered and NAND gate.

5) Design the 74LS160 to count in the sequence $(4,5,6,7,8)$. (explain your idea and show schematic diagram).

*Note:* draw any basic gate used individually without IC.