The 555 Timer

The 555 Timer is a versatile and widely used device which can be configured as a mono-stable One-Shot or as an Astable multivibrator. An Astable multivibrator is known as an Oscillator which does not have any stable state. Therefore it continuously changes from one unstable state to the other without any external trigger.

Timing Problem in flip-flop circuits

In synchronous digital circuits the output of one flip-flop is connected to the input of a second flip-flop, either directly or through logic gates. Both the flip-flops are triggered through a common clock signal connected to the clock input of both the flip-flops. This leads to a potential timing problem as shown in figure 26.1.

Assume the initial outputs of flip-flop 1 and 2 are at logic high and low respectively. When there is a high to low clock transition \( t_1 \), the output of flip-flop 1 toggles to logic low. The high to low clock transition at the clock input of Flip-flop 2 also occurs at the same instant \( t_1 \). During the interval \( t_1 \) and \( t_2 \) the output of flip-flop is changing from logic high to logic low and will go to logic low after a propagation delay.
The input to flip-flop 2 is changing from logic high to low during the time interval $t_1$ and $t_2$. The input to flip-flop 2 should be held stable for a minimum hold time requirement $t_{H}$. If the input is not held stable for $t_{H}$ interval the output can not be guaranteed to be logic high. Output of flip-flop 2 will be set to logic high state if $t_{PHL}$ time of flip-flop 1 is more than the $t_{H}$ of flip-flop 2. Practically flip-flops have hold times that are 5 nsec or less, most have $t_{H} = 0$. Therefore, flip-flop circuits such as the one shown connected in the diagram work reliably.

Clock Skew

One of the most common problems in synchronous circuits is ‘Clock Skew’. One type of Clock Skew occurs when the same clock signal arrives at different times at different clock inputs to propagation delay, which causes different flip-flops to change states asynchronously leading to unpredictable outputs. Figure 26.2
In the circuit diagram both the flip-flops are connected to the same clock signal. However, the clock signal to the second flip-flop is delayed by the NAND and NOT gates. On a high to low clock transition both the flip-flops change their output states assuming that the initial output state of each flip-flop is logic low. The Clock Skew is the delay in the two clock signals by a time interval $t_1 \ t_2 \ or \ t_3 \ t_4$. At the high to low transition of clock 1 signal the output of F1 toggles from logic low to logic high after a propagation delay of $t_{PLH}$. If the propagation delay of F1 is less than the clock skew then at the high to low clock transition of clock 2 the J input of flip-flop is set to logic high and at the clock transition the output F2 is set to logic high. If the propagation delay $t_{PLH}$ of F1 is of a longer duration than the Clock Skew, the J input of the flip-flop is at logic low at the high to low transition of clock 2 the output of F2 remains unchanged.

Timing problems occurring due to clock skew are intermittent in nature and therefore are difficult to detect. The clock skew can vary with changes in temperature, power supply voltages, length of connections and loading effects. Problems caused due to clock skew can be eliminated by equalizing clock delays to different parts of the circuit.

**Race Conditions**

Race conditions are said to occur when multiple internal variables change due to change in one input variable. Depending upon the sequence in which the internal variables change, the circuit output operates erratically. Figure 26.3. In the timing diagram shown, if the Q and $\overline{Q}$ output high to low transitions are slightly delayed, they coincide with the clock low to high transitions resulting in short duration pulses which are difficult to detect. The glitches due to race condition can be avoided by using a negative-edge triggered flip-flop instead of the positive-edge-triggered flip-flop used.

![J-K flip-flop circuit with potential race condition](image)
Figure 26.3b  Timing Diagram showing glitches due to race conditions

Figure 26.3c  Timing Diagram of negative-edge triggered flip-flop avoiding glitches

Clock1
Counters

Counter circuits based on flip-flops are widely used in Digital Systems. Besides counting, these counters are used as frequency dividers and with minor changes in the circuit they are used as shift registers. Counters are classified as Asynchronous and Synchronous counters. Asynchronous counters as the name indicates are not triggered simultaneously. The multiple flip-flops that are connected together to form a counter circuit do not receive the triggering clock signal simultaneously. The flip-flop that represents the least significant count bit of the n-bit counter is connected to the clock signal, the remaining flip-flops receive their clock signals form the outputs of the preceding flip-flops connected in the counter circuit. The clock signal thus ripples through successive flip-flops. Synchronous counters on the other hand have all the clock inputs of the multiple flip-flops connected to a common clock signal. All the flip-flops in a Synchronous counter receive clock signals simultaneously.

Asynchronous and Synchronous are further classified as up counters or down counters depending upon the sequence in which they count. They are further classified in terms of the number of states or the range of numbers to which the counters can count.

Asynchronous Counters (Ripple Counters)

Asynchronous counters are implemented by connecting together multiple flip-flops together. The triggering clock signal is connected to the clock input of the first flip-flop. The clock inputs of the remaining flip-flops are connected to the Q or Q output of the previous flip-flop. On a clock transition at the clock input of the first flip-flop the output state of the flip-flop changes. With the transition in the output state of the first flip-flop, there is also a transition at the clock input to the second flip-flop as the output of the first flip-flop is connected to the clock input of the second flip-flop. Due to the clock transition the second flip-flop changes its output state. The change in the output state of the second flip-flop occurs after the first flip-flop changes its state. Similarly, the last flip-flop connected in the counter circuit changes its output state after the output of the flip-flop connected to its clock input has changed it state. The outputs of the flip-flops change in a sequence as the clock signal propagates through the flip-flops as they change their output states one after the other. The Asynchronous counters are also known as Ripple Counters due to the rippling effect of the clock signal.

A three-bit Asynchronous counter circuit is shown in Figure 26.4. In the circuit diagram shown the Q output of each is connected to the clock input of the next flip-flop. The J-K inputs of each of the three flip-flop are connected to logic high allowing the flip-flop to toggle their output state on a high to low transition at their clock input.

The output state of the first flip-flop toggles at every positive to negative clock transition in intervals t₁ to t₈. The output F₁ of the second flip-flop toggles at intervals t₂, t₄, t₆ and t₈ on every high to low transition of the output F₀. The output F₂ toggles its output state at intervals t₄ and t₈ on a high to low transition of the flip-flop output F₁.
Figure 26.4a 3-bit Asynchronous Up-Counter

Figure 26.4b Timing Diagram of a 3-bit Asynchronous Up-Counter

<table>
<thead>
<tr>
<th>Input Clock Pulses</th>
<th>Output F₂</th>
<th>Output F₁</th>
<th>Output F₀</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 26.1 Output State of a 3-bit Asynchronous Up-Counter
Propagation Delay

The timing diagram shown in figure 26.4b doesn’t take into account the propagation delay that occurs between each clock input and the corresponding toggling output. The timing diagram which takes into account the propagation delay is shown in figure 26.5. At time interval $t_4$ on a clock transition the output $F_0$ toggles to a new state after a delay determined by $t_{PHL}$ propagation delay of the first flip-flop. At interval $t_5$ on the high to low transition of the $F_0$ output, the output $F_1$ toggles to a new state after a propagation delay $t_{PHL}$ of the second flip-flop. Finally, at interval $t_6$ the transition in $F_1$ output toggles the output $F_2$ of the third flip-flop. The output $F_2$ becomes stable after a propagation delay $t_{PLH}$ of the third flip-flop. The propagation delay of each of the three flip-flop adds up to delay the output $F_2$ by three propagation delays with respect to the clock transition at interval $t_4$. If the counter circuit is extended by adding more flip-flops, then the output of the last flip-flop might exceed the clock period of the clock which causes timing problems. The Asynchronous counters can not work at high clock frequencies and cause problems with decoding circuits.

![Figure 26.5 Timing Diagram of a 3-bit Asynchronous with propagation delay](image)

The timing diagram of the 3-bit counter circuit using a clock of a higher frequency is shown in Figure 26.6a. At interval $t_4$, the negative clock transition toggles the $F_0$ output to logic low at interval $t_A$ after a propagation delay of $t_{PHL}$. The negative transition of $F_0$ at $t_A$ toggles the $F_1$ output to logic low at interval $t_5$ after a propagation delay of $t_{PHL}$. Finally, the $F_2$ output is toggled to logic high at interval $t_B$ after a delay of $t_{PLH}$ after the clock ($F_1$) transition at interval $t_5$. The output states of the counter at intervals $t_1$ to $t_7$ are shown in table 26.2. The output at interval $t_5$ should be 100 instead of 010.
Table 26.2 Output of a 3-bit Asynchronous Up-Counter with high frequency clock

### Mod-n Counters

The term Mod represents the Modulus of the counter which is the total number of unique states through which the counter will sequence through. A 3-bit Asynchronous counter can count up from 0 to 7 or count down from 7 to 0. The 3-bit counter has 8 different states represented by the 8 outputs 0 to 7. The counter states or the range of numbers of a counter is determined by the formula $2^n$, where $n$ represents the number of flip-flops. Therefore, a Mod-8 counter implemented using three flip-flops $2^3$ has 8 output states.

Counter can also be designed to have less number of states than $2^n$. The resulting sequence is called a truncated sequence. The counter therefore counts up to the truncated sequence. Designing a truncated sequence counter is very simple. When the counter counts up to the intended sequence it is reset to the initial count value 0. The counter is reset to the initial count value by activating the Clear asynchronous inputs. The clears input is activated by the counter through a combinational circuit that activates its output.
when the appropriate count sequence is reached. The Mod-6 counter is shown in figure 26.7.

![Mod-6 Counter](image1)

**Figure 26.7a  Mod-6 Counter**

The counter counts from state 000 to 101. At interval $t_6$ the counter counts to 110. The outputs $F_1$ and $F_2$ of the counter are connected to the inputs of a 2-input NAND gate, which sets its output to logic zero when both its inputs become logic 1 at interval $t_6$. The output of the NAND gate is connected to the three active-low asynchronous Clear input of the three flip-flops which are set to low by the NAND gate. Therefore the counter is immediately reset to state 000 from where it proceeds to sequence through the count values. The Mod number of the counter also determines the frequency at the output of the counter. The output at $F_2$ has a frequency which is $1/6^{th}$ of the input clock frequency.
Thus Mod-n counters can be designed to generate $1/n$th frequency signal with respect to the input clock signal.

Mod-10 Counter (Decade Counter)

A decade counter uses four flip-flops to implement the circuit which counts up to 10 unique states (0000 to 1001). The counter is reset when it counts to the next state 1010. The frequency of the output signal is $1/10$th the input clock frequency. Figure 26.8.

\[ \text{J-K flip-flop 1} \]
\[ \text{J-K flip-flop 2} \]
\[ \text{J-K flip-flop 3} \]
\[ \text{J-K flip-flop 4} \]

Figure 26.8a  Asynchronous Decade Counter

The output $F_1$ and $F_3$ are connected through a NAND gate to the active-low clear inputs of all the four flip-flops. The counter counts from 0000 to 1001 (ten output states),
when it counts to 1010, the output of the NAND gate is set to logic low which resets all the four flip-flops to state 0000.

Integrated Circuit Asynchronous Counters

Asynchronous Counters are available in Integrated Circuit form. The 74LS93A is a 4-bit Asynchronous Counter. The counter has two separate clock inputs CLK A and CLK B connected to the clock input of the first and second flip-flop respectively. The second, third and fourth flip-flops are internally connected as a ripple 3-bit counter. The counter also has two inputs pins connected to the inputs of a 2-input NAND (internal) gate, the output off which is connected to the clear inputs of all the four flip-flops. The counter provides four outputs, one form each flip-flop. Figure 26.9

![Internal circuit diagram of the 74LS93A Counter](image)

The 74LS93A can be configured as MOD-16 counter by connecting CLK B input pin to the Q0 output pin of the IC. RO 1 and RO 2 are connected to logic low. A Decade counter can be implemented by connecting CLK B input to the Q0 and Q1 and Q3 outputs to RO 1 and RO 2 respectively. Figure 26.10 Two 74LS93As can be cascaded together to form a larger counter. A MOD-50 counter is implemented using two 74LS93A ICs. Figure 26.11

![74LS93A connected as MOD-16 Counter](image)
In the circuit diagram two 74LS93As are connected together to form a frequency divider which divides the input frequency by 50. The first 74LS93A is connected to divide the input frequency by 10. The Q₃ output of the first 74LS93A is connected to the CLKB input of the second 74LS93A. The second 74LS93A is connected to divide the input frequency at CLKB by 5. The Q₃ output of the second 74LS93A therefore provides an output which is 1/5₀th of the clock applied at the CLKA input of the first 74LS93A. The second 74LS93A requires the use of only three flip-flops, therefore the first flip-flop with clock input CLKA is left unconnected.