Chapter - 5

**FLIP-FLOPS AND SIMPLE FLIP-FLOP APPLICATIONS**

**Introduction :**

Logic circuit is divided into two types.

1. Combinational Logic Circuit
2. Sequential Logic Circuit

**Definition :**

1. Combinational Logic Circuit :
   The circuit in which outputs depends on only present value of inputs. So it is possible to describe each output as function of inputs by using Boolean expression. No memory element involved. No clock input. Circuit is implemented by using logic gates. The propagation delay depends on, delay of logic gates. Examples of combinational logic circuits are : full adder, subtractor, decoder, codeconverter, multiplexers etc.

   ![Combinational Logic Circuit Diagram](image)

2. Sequential Circuits :
   Sequential Circuit is the logic circuit in which output depends on present value of inputs at that instant and past history of circuit i.e. previous output. The past output is stored by using memory device. The internal data stored in circuit is called as state. The clock is required for synchronization. The delay depends on propagation delay of circuit and clock frequency. The examples are flip-flops, registers, counters etc.

   ![Sequential Circuit Diagram](image)
Basic Bistable element.
- Flip-Flop is Bistable element.
- It consists of two cross-coupled NOT Gates.
- It has two stable states.
- \(Q\) and \(\bar{Q}\) are two outputs complement of each other.
- The data stored 1 or 0 in basic bistable element is the state of flip-flop.
- 1 – State is set condition for flip-flop.
- 0 – State is reset/clear for flip-flop.
- It stores 1 or 0 state as long power is ON.

Latches:
S-R Latch: Set-reset Flip-Flop
- Latch is a storage device by using Flip-Flop.
- Latch can be controlled by direct inputs.
- Latch outputs can be controlled by clock or enable input.
- \(Q\) and \(\bar{Q}\) are present state for output.
- \(Q^+\) and \(\bar{Q}^+\) are next states for output.
- The function table/Truth table gives relation between inputs and outputs.
- The \(S=R=1\) condition is not allowed in SR FF as output is unpredictable.
Application of SR Latch:

- A switch debouncer

- Bouncing problem with Push button switch.
- Debouncing action.
- SR Flip-Flop as switch debouncer.
Gated SR Latch:

- Enable input C is clock input.
- C=1, Output changes as per input condition.
- C=0, No change of state.
- S=1, R=0 is set condition for Flip-flop.
- S=0, R=1 is reset condition for Flip-flop.
- S=R=1 is ambiguous state, not allowed.

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>S R C</td>
<td>Q bar Q*</td>
</tr>
<tr>
<td>0 0 1</td>
<td>Q Q</td>
</tr>
<tr>
<td>0 1 1</td>
<td>0 1</td>
</tr>
<tr>
<td>1 0 1</td>
<td>1 0</td>
</tr>
<tr>
<td>1 1 1</td>
<td>1* 1*</td>
</tr>
<tr>
<td>X X 0</td>
<td>Q Q</td>
</tr>
</tbody>
</table>

*Unpredictable behavior will result if S and R return to 0 simultaneously or C returns to 0 while S and R are 1*
JK Flip-Flop by using SR Flip-Flop

In SR FF, S=R=1 condition is not allowed.

- JK FF is modified version of SR FF.
- Due to feedback from output to input AND Gate J=K=1 is toggle condition for JK FF.
- The output is complement of the previous output.
- This condition is used in counters.
- T-FF is modified version of JK FF in which T=J=K=1.

Function Table

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>J</td>
<td>K</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

In SR FF, S=R=1 condition is not allowed.
Gated D Latch:

- D Flip-Flop is Data Flip-Flop.
- D Flip-Flop stores 1 or 0.
- R input is complement of S.
- Only one D input is present.
- D Flip-Flop is a storage device used in register.
Master slave SR Flip-Flop

- Two SR Flip-Flop, 1\textsuperscript{st} is Master and 2\textsuperscript{nd} is slave.
- Master Flip-Flop is positive edge triggered.
- Slave Flip-Flop is negative edge triggered.
- Slave follows master output.
- The output is delayed.
Master slave JK Flip-Flop

- In SR Flip-Flop the input combination S=R=1 is not allowed.
- JK FF is modified version of SR FF.
- Due to feedback from slave FF output to master, J=K=1 is allowed.
- J=K=1, toggle, action in FF.
- This finds application in counter.
Positive Edge Triggered D Flip-Flop

- When C=0, the output of AND Gate 2 & 3 is equal to 1.
  \[ S = R = 1, \text{ No Change of State} \]

- If C=1, D=1, the output of AND Gate 2 is 0 and 3 is 1.
  \[ S = 0, R = 1, \quad Q = 1 \text{ and } Q' = 0 \]
REGISTERS

- Register is a group of Flip-Flops.
- It stores binary information 0 or 1.
- It is capable of moving data left or right with clock pulse.
- Registers are classified as
  - Serial-in Serial-Out
  - Serial-in parallel Out
  - Parallel-in Serial-Out
  - Parallel-in parallel Out

Fig. : Serial-In, Serial-Out Unidirectional Shift Register

Fig. : Serial-In, Parallel-Out Unidirectional Shift Register
Parallel-in Unidirectional Shift Register

- Parallel input data is applied at I_AI_BI_CI_D.
- Parallel output Q_AQ_BQ_CQ_D.
- Serial input data is applied to A FF.
- Serial output data is at output of D FF.
- \( \bar{L}/\text{Shift} \) is common control input.
- \( \bar{L}/S = 0 \), Loads parallel data into register.
- \( \bar{L}/S = 1 \), shifts the data in one direction.

Fig. : Parallel-in Unidirectional Shift Register
Universal Shift Register

- Bidirectional Shifting.
- Parallel Input Loading.
- Serial-Input and Serial-Output.
- Parallel-Input and Serial-Output.
- Common Reset Input.
- 4:1 Multiplexer is used to select register operation.
**COUNTERS**

- Counter is a register which counts the sequence in binary form.
- The state of counter changes with application of clock pulse.
- The counter is binary or non-binary.
- The total no. of states in counter is called as modulus.
- If counter is modulus-n, then it has n different states.
- State diagram of counter is a pictorial representation of counter states directed by arrows in graph.

![Fig. State diagram of mod-8 counter](image-url)
4-bit Binary Ripple Counter:

- All Flip-Flops are in toggle mode.
- The clock input is applied.
- Count enable = 1.
- Counter counts from 0000 to 1111.
Synchronous Binary Counter:

- The clock input is common to all Flip-Flops.
- The T input is function of the output of previous flip-flop.
- Extra combination circuit is required for flip-flop input.
Counters Based on Shift Register

- The output of LSB FF is connected as D input to MSB FF.
- This is commonly called as Ring Counter or Circular Counter.
- The data is shifted to right with each clock pulse.
- This counter has four different states.
- This can be extended to any no. of bits.

Twisted Ring Counter or Johnson Counter

- The complement output of LSB FF is connected as D input to MSB FF.
- This is commonly called as Johnson Counter.
- The data is shifted to right with each clock pulse.
- This counter has eight different states.
- This can be extended to any no. of bits.
Mod-7 Twisted Ring Counter

- The D input to MSB FF is $\overline{Q}_D, \overline{Q}_C$
- The counter follows seven different states with application of clock input.
- By changing feedback different counters can be obtained.

Design Procedure for Synchronous Counter

- The clock input is common to all Flip-Flops.
- Any Flip-Flop can be used.
- For mod-n counter 0 to n-1 are counter states.
- The excitation table is written considering the present state and next state of counter.
- The flip-flop inputs are obtained from characteristic equation.
- By using flip-flops and logic gate the implementation of synchronous counter is obtained.
**Difference between Asynchronous and Synchronous Counter:**

<table>
<thead>
<tr>
<th>Asynchronous Counter</th>
<th>Synchronous Counter</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Clock input is applied to LSB FF. The output of first FF is connected as clock to next FF.</td>
<td>1. Clock input is common to all FF.</td>
</tr>
<tr>
<td>2. All Flip-Flops are toggle FF.</td>
<td>2. Any FF can be used.</td>
</tr>
<tr>
<td>3. Speed depends on no. of FF used for n bit.</td>
<td>3. Speed is independent of no. of FF used.</td>
</tr>
<tr>
<td>$t_{max} = \frac{1}{n \times t_p}$</td>
<td>$t_{max} = \frac{1}{t_p}$</td>
</tr>
<tr>
<td>4. No extra Logic Gates are required.</td>
<td>4. Logic Gates are required based on design.</td>
</tr>
<tr>
<td>5. Cost is less.</td>
<td>5. Cost is more.</td>
</tr>
</tbody>
</table>
