Study of Various Flip-Flop Circuits

Gayatri P
2nd year, Integrated M.Sc. Physics
Roll No.: 2211185
(Dated: April 17, 2024)

In this experiment we contruct and study sequential logic circuits using various kinds of flip-flop circuits. We also discuss certain practicalities and limitations of each one, and how to overcome them.

I. OBJECTIVE

To construct and study the operations of the following flip-flop circuits:

1. RS and Clocked RS Flip-Flop
2. D Flip-Flop
3. JK and Master-Slave JK Flip-Flop
4. T Flip-Flop

II. THEORY

Previously, we have worked with **combinational circuits**, for which the output is determined completely by the existing state of the inputs. There is no "memory," in these circuits. Circuits that store some history makes it possible to construct counters, arithmetic accumulators, and circuits that generally do one thing after another. Such circuits that remember their current output or state are often called **sequential logic circuits**.

The basic principle of memory is the concept of **feedback**.

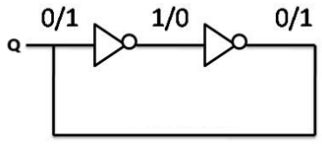

FIG. 1: Simplest realization of feedback circuit

In Fig. 1, if Q happens to be 1 (or 0), it will always be 1 (or 0). While this circuit is of not much use, we use this principle in realizing the the most basic form of a sequential circuit that is **flip-flop**. A flip-flop circuit has two outputs, one for the normal value and one for the complement value of the stored bit.

RS Flip-Flop

The simplest form of a flip-flop circuit, it can be constructed using either `NAND` or `NOR` gates.

An RS flip-flop circuit using `NOR` gates is shown below. The inputs R and S refers to `SET` and `RESET` respectively.

- Assume that $S = 1$ and $R = 0$. The output of the bottom `NOR` gate is $Q' = 0$. So, both inputs to the top NOR gate, $Q = 1$. Hence, the input combination leads to the flip-flop being **set** to $Q = 1$ and $Q' = 0$.
- Similarly, if $S = 0$ and $R = 1$, $Q = 0$ and $Q' = 1$, i.e. the flip-flop is **reset**.
- Assume the flip-flop is in the set state. Now if one changes S to 0, the output still remains the same ($Q = 1$ and $Q' = 0$). Similary, if the flip-flop is in the reset state and we change R to 0, the output still remains the same ($Q = 0$ and $Q' = 1$). This means in the state $S = 0, R = 0$, the ciruit retains its previous output, also knwon as the **hold** state.
- When both $R = 1$ and $S = 1$, both outputs are no longer compliments of each other. Since it is impossible to predict which output will go to 1, this is an invalid state making it one of the main disadvantages of the RS flip-flop.

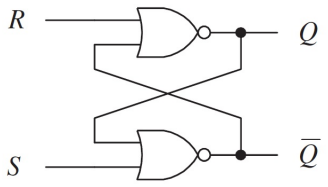

FIG. 2: Circuit diagram of an RS flip-flop

R	S	Q	Q'	Remark
0	0	Q	Q'	`HOLD`
0	1	1	0	`SET`
1	0	0	1	`RESET`
1	1	-	-	Invalid

TABLE I: Characteristic Table for an RS Flip-Flop

Switch Debouncing: An useful example using this flip-flop is the debounce circuit. Suppose a piece of electronics is to change state under the action of a mechanical switch. When this switch is moved from position S to R ($S = 0, R = 1$), When the switch is closed, the two contacts actually separate and reconnect, typically 100s of times over a period of about a millisecond. It is desirable that the electronics should respond to the first contact and then remain stable, rather than switching back and forth as the circuit makes and breaks.

This is achieved by RS flip-flop which is reset to $Q = 0$ by the first signal $R = 1$ and remains in a fixed state until

the switch is moved back to position S, when the signal $S = 1$ sets the flip-flop to $Q = 1$.

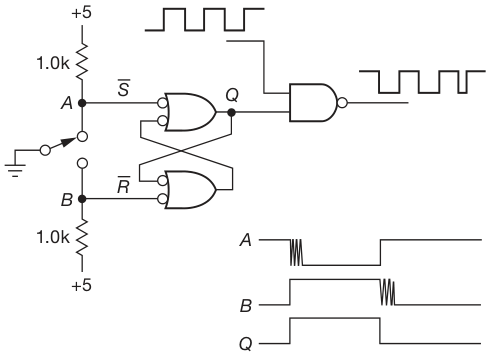

FIG. 3: An example of an SR flip-flop switch debouncer.

Gated or Clocked RS Flip-flop

By connecting an AND gate in series with each input terminal of the RS NOR Flip-flop, we can construct a Gated RS Flip-flop. This extra conditional input is called an *Enable* (EN) input. When the $EN = 0$, the outputs of the two AND gates are also at 0, regardless of the inputs S and R, hence latching the two outputs Q and Q' into their last known state. When the $EN = 1$, the circuit acts like a normal RS bistable flip-flop.

EN input can also be connected to a clock timing signal adding clock synchronisation to the flip-flop creating what is sometimes called a *Clocked RS Flip-flop*.

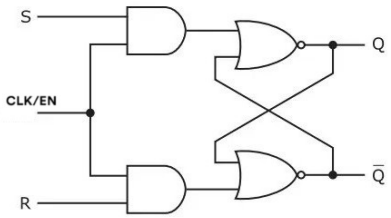

FIG. 4: Circuit diagram of a gated RS flip-flop

Q_N	R	S	Q_{N+1}	Remark
0	0	0	0	HOLD
0	0	1	0	
0	1	0	1	
0	1	1	-	Indeterminate
1	0	0	1	HOLD
1	0	1	0	
1	1	0	1	
1	1	1	-	Indeterminate

TABLE II: Characteristic Table for a Gated RS Flip-flop

A. D Flip-flop

The D flip-flop has only a single data input D. This D input and its inverse are connected to S and R inputs respectively. Thus, this eliminates undefined state of $R = S = 1$ seen in case of RS flip-flops. A D-flip flop has a second input called Enable, EN, to allow the flip-flop to be in a holding state, as seen in the circuit diagram.

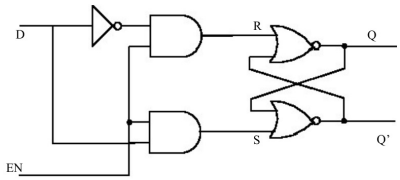

FIG. 5: Circuit diagram of a D flip-flop

- When $EN = 1$, $S = D$ and $R = D'$. Hence the output Q follows D.
- When $EN = 0$, irrespective of D, the most recent state is held.

Q_N	D	Q_{N+1}
0	0	0
0	1	1
1	0	0
1	1	1

TABLE III: Characteristic Table for a D Flip-flop, with $EN = 1$ (hence no HOLD state).

JK Flip-flop

The JK flip flop, named after Jack Kilby who invented it is the most commonly used flip-flop. It is basically a gated SR flip-flop with the addition of a clock input circuitry that prevents the illegal or invalid output condition that can occur when both inputs $S = R = 1$. Due to this additional clocked input, a JK flip-flop has four possible output combinations, 1, 0, *no change* and *toggle*.

Note that in the following circuit diagram NAND gates are used instead of NOR gates.

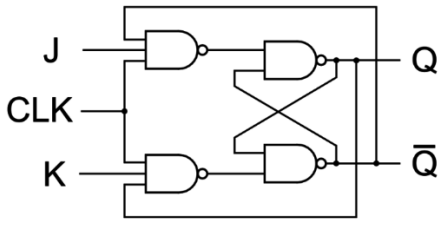

FIG. 6: Circuit diagram of a JK flip-flop

The two 2-input AND gates of the gated SR bistable have now been replaced by two 3-input NAND gates with

the third input of each gate connected to the outputs at Q and Q. This cross coupling allows the previously invalid condition of $S = R = 1$ state to be used to produce toggle action as the two inputs are now interlocked. We can summarize its operation as follows.

- Assume the clock signal is 1. If $J = 1$ and $K = 0$, it behaves as a typical SR latch, i.e. $Q = 1$ and $Q' = 0$. Similarly for $J = 0$ and $K = 1$, $Q = 0$ and $Q' = 1$.
- If both $J = K = 0$, then it remains in the same state as it was before i.e. the `HOLD` state.
- If both $J = K = 1$, the flip-flop changes state whenever a clock pulse occurs; i.e., the clock pulse toggles the flip-flop again and again until the Clk goes to 0. This is called is *race-around condition* (Fig. 7) and is undesirable, which is eliminated by an improvised form of this flip-flop as discussed next.

Q_N	J	K	Q_{N+1}	Remark
0	0	0	0	`HOLD`
0	0	1	0	
0	1	0	1	
0	1	1	1	`TOGGLE`
1	0	0	1	`HOLD`
1	0	1	0	
1	1	0	1	
1	1	1	0	`TOGGLE`

TABLE IV: Characteristic Table for a JK Flip-flop

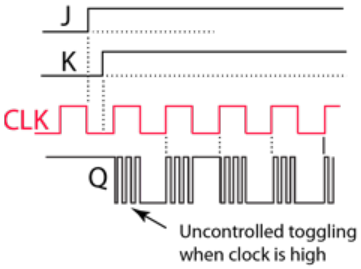

FIG. 7: Diagram depicting JK racing when the clock is high

Master-Slave JK Flip-flop

To avoid racing in the JK flip-flop, the timing pulse period must be kept as short as possible, which is not very practical with modern TTL IC's. A Master-slave JK solves this problem by using two SR flip-flops connected in series – the *master circuit*, which triggers the leading edge of the clock pulse and the *slave circuit*, which triggers the falling edge of the clock pulse. Hence the uncontrolled toggling is suppressed by as the transmission of the J value to the output is delayed by half a clock cycle and not immediately fed back to the input side.

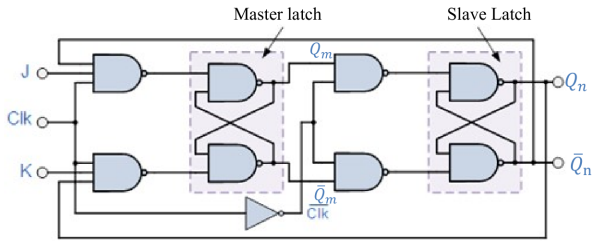

FIG. 8: Circuit diagram of a Master-slave JK flip-flop

When a clock pulse enables the master flip-flop, it disables the slave flip-flop (as its clock signal is inverted). When the clock changes state again (i.e., on its falling edge) the output of the master latch is transferred to the slave latch. Again, toggling is accomplished by the connection of the output with the input AND gates.

CP	J	K	Q_m	Q'_m	Q_n	Q'_n
$0 \to 1$	0	0	Hold		Hold	
$1 \to 0$	0	0	Hold		Hold	
$0 \to 1$	0	1	0	1	Hold	
$1 \to 0$	0	1	Hold		0	1
$0 \to 1$	1	0	1	0	Hold	
$1 \to 0$	1	0	Hold		1	0
$0 \to 1$	1	1	Toggle		Hold	
$1 \to 0$	1	1	Hold		Toggle	

TABLE V: Characteristic Table for a Master-slave JK Flip-flop

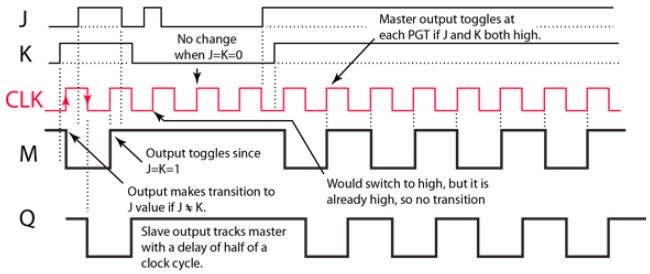

FIG. 9: The output changes once in a clock cycle in case of a master-slave JK flip-flop. This solves the racing problem in a single stage JK flip-flop.

T Flip-flop

T or Toggle flip-flop is a modification on the master-slave JK flip-flop where it only performs toggling action. Here, we make $J = K = 1$.

- When Clk $= 0$, the state will be stored (`HOLD`) regardless of the value of T.
- When Clk $= 1$ and $T = 0$, the previous state is stored, i.e. `HOLD` state.

- When Clk = 1 and $T = 1$, we will have toggle action.

Q_N	T	Q_{N+1}	Remark
0	0	0	HOLD
0	1	1	TOGGLE
1	0	1	HOLD
1	1	0	TOGGLE

TABLE VI: Characteristic Table for a T Flip-flop, where Clk = 1

III. APPARATUS

1. ICs (NOR-7402, AND(2-input)-7408, NAND(3-input)-7410, NAND-7400, NOT-7404)
2. Resistors (1 kΩ)
3. DC Power Supply (5V)
4. LEDs
5. Connecting Wires
6. Multimeters

IV. OBSERVATIONS

1. **RS Flip-flop using NOR gates**
Refer to Fig. 2.

R	S	Q	Q'	Remark
0	1	1	0	SET
0	0	1	0	HOLD
1	0	0	1	RESET
0	0	0	1	HOLD
1	1	0	0	Invalid

TABLE VII: Observed characteristic Table for an RS Flip-Flop

2. **RS Flip-flop using NAND gates**

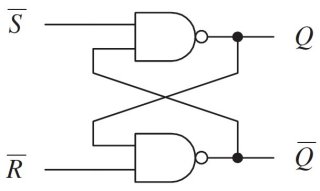

FIG. 10: Circuit diagram of an RS flip-flop using NAND gates. Q and Q' are instead configured at S' and R' respectively, unlike Fig 2.

S'	R'	Q	Q'	Remark
0	1	1	0	SET
1	1	1	0	HOLD
1	0	0	1	RESET
1	1	0	1	HOLD
0	0	0	0	Invalid

TABLE VIII: Observed characteristic Table for an RS Flip-Flop using NAND gates.

Notice how the indeterminate state is defined when $S = R = 0$ and the Hold state is when $S = R = 1$, due to the fundamental difference between NAND and NOR properties.

3. **Gated RS Flip-flop**
Refer to Fig. 4.

Q_N	R	S	Q_{N+1}	Remark
0	1	0	0	RESET
0	0	0	0	HOLD
0	0	1	1	SET
1	0	0	1	HOLD
1	0	1	1	SET
1	1	0	0	RESET
0	1	1	0	Indeterminate
1	1	1	0	Indeterminate

TABLE IX: Observed characteristic Table for a gated RS Flip-Flop

4. **D Flip-flop**
Refer to Fig. 5.

Q_N	D	Q_{N+1}
0	1	1
1	0	0
0	1	1
1	1	1

TABLE X: Observed characteristic Table for a gated RS Flip-Flop, with Clk = 1

5. **JK Flip-flop**
Refer to Fig. 6.

Q_N	J	K	Q_{N+1}	Remark
0	0	0	0	HOLD
0	0	1	0	
0	1	0	1	
1	0	1	0	
1	1	0	1	
1	0	0	1	HOLD
1	1	1	0	TOGGLE
0	1	1	1	TOGGLE

TABLE XI: Observed characteristic Table for a JK Flip-Flop

6. **Master-Slave JK Flip-flop**
   Refer to Fig. 8.

CP	J	K	Q_m	Q'_m	Q_n	Q'_n
$0 \to 1$	0	1	0	1	Hold	
$1 \to 0$	0	1	Hold		0	1
$0 \to 1$	1	0	1	0	Hold	
$1 \to 0$	1	0	Hold		1	0
$0 \to 1$	0	0	Hold		Hold	
$1 \to 0$	0	0	Hold		Hold	
$0 \to 1$	1	1	Toggle		Hold	
$1 \to 0$	1	1	Hold		Toggle	

TABLE XII: Observed characteristic Table for a Master-Slave JK Flip-Flop

7. **JK Flip-flop**
   Refer to Fig. 8, where J & K are set to 1.

Q_N	T	Q_{N+1}	Remark
0	0	0	`HOLD`
0	1	1	`TOGGLE`
1	0	1	`HOLD`
1	1	0	`TOGGLE`

TABLE XIII: Observed characteristic Table for a T Flip-Flop

V. DISCUSSION & CONCLUSION

In this experiment we contructed and studied sequential logic circuits using various kinds of flip-flop circuits. We discussed simple RS flip-flop using both NOR and NAND gates, and its practical application in switch debouncing circuits. We discussed RS flip-flops with a clock signal as well as D flip-flops which are able to remove the indeterminate state in RS flip-flops.

Furthermore, we dicussed JK flip-flops and its purpose, including toggle action and how to counteract the racing problem using a master-slave flip-flop. We also discussed a modified version of the master-slave JK flip-flop, the T flip-flop, which only performs `TOGGLE` and `HOLD` actions.

VI. PRECAUTIONS

1. Make sure the connections are proper before switching on the circuit.
2. Make sure to connect a resistor parallel to the LED as to not burn out the LED.

VII. APPLICATIONS

Flip-flops play a critical role in computer electronics by serving as memory elements, storing state information, ensuring clock synchronization, enabling digital counting, and facilitating control logic. They are essential for data storage, sequencing, coordination, and control within a computer system.

[1] SPS, *Study of Various Flip-Flop Circuits*, NISER (2023).

[2] P. Horowitz and W. Hill, *The art of electronics* (Cambridge University Press, 2015).