1.1. Introduction

A latch or flip-flop is a bistable circuit that is most often used in applications that require data storage. Its chief characteristic is that the output is not dependent solely on the present state of the input but also on the preceding output state. A bistable circuit has two complementary outputs that can assume either of the two logic levels 0 or 1.

There are several common types of latches and flip-flops. Latches often have no dedicated input for the clock signal. They can be combined to implement level-triggered and edge-triggered flip-flops. Flip-flops can be triggered by one of the levels or one of the edges of a clock signal (or a digital signal).

1.2. General overview

A simple latch can be implemented using two NOR or two NAND logic gates.

A NOR gate based latch with initial conditions specified is represented in Figure 1.1(a). The characteristic equation for each of the outputs is determined by assuming that the logic gates have different propagation times\(^1\) and this may be modeled as for a delay, \(\Delta\), between a signal that becomes available at the output and the feedback signal applied to the input. In this way, the logic circuit of the latch, as illustrated in Figure 1.1(b), may be transformed as shown in Figures 1.1(c) and 1.1(d).

---

1 Propagation delays in logic gates are assumed to take the form 1 and 1 + \(\Delta\), respectively.
Referring to Figure 1.1(c), we can write:

\[ X = B + Y \quad \text{[1.1]} \]
\[ Y^+ = \overline{A + X} \quad \text{[1.2]} \]

Substituting [1.1] into [1.2] yields:

\[ Y^+ = \overline{A + B + Y} \quad \text{[1.3]} \]
\[ = \overline{A} \cdot \overline{B + Y} \]
\[ = \overline{A} \cdot (B + Y) \]
\[ = \overline{A} \cdot B + \overline{A} \cdot Y \quad \text{[1.4]} \]

Similarly, the circuit shown in Figure 1.1(d) can be characterized using the following logic equations:

\[ X^+ = B + Y \quad \text{[1.5]} \]
\[ Y = \overline{A + X} \quad \text{[1.6]} \]

By substituting [1.5] into [1.6], we have:

\[ X^+ = \overline{B + A + X} \quad \text{[1.7]} \]
\[ = \overline{B} \cdot \overline{A + X} \]
\[ = \overline{B} \cdot (A + X) \]
\[ = A \cdot \overline{B} + \overline{B} \cdot X \quad \text{[1.8]} \]
The characteristic equations of the NOR gate based latch are, thus, given by:

\[ X^+ = A \cdot \overline{B} + \overline{B} \cdot X \] \hspace{1cm} [1.9]

and

\[ Y^+ = \overline{A} \cdot B + \overline{A} \cdot Y \] \hspace{1cm} [1.10]

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>X</th>
<th>X⁺</th>
<th>Y⁺</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Table 1.1. State table of the NOR gate based latch**

For each output, the next state, \( X^+ \) or \( Y^+ \), depends on the present state, \( X \) or \( Y \). In addition to the characteristic equations, the initial conditions must be specified to determine the operation of the latch. Table 1.1 gives the state table for the latch.

It must be noted that the two signals, \( X^+ \) and \( Y^+ \), are complementary except when both inputs, \( A \) and \( B \), are set to 1.

Additionally, if the inputs \( A \) and \( B \) are simultaneously set to 0, the outputs can no longer be defined in a unique manner as the characteristic equations are verified by \((X, Y) = (1, 0)\) or by \((X, Y) = (0, 1)\). It is therefore impossible to predict the combination of the states held by the outputs.

In practice, sequential circuits are most often made to operate in the *fundamental mode*. This means that only one input can change states at any time. On the other hand, because of the difference in propagation delays between the logic gates, it is impossible to guarantee a simultaneous change in the state of two variables. Thus, the outputs of the latch are defined by \((X, Y) = (0, 1)\) when \( A \) is first set to 0 or by \((X, Y) = (1, 0)\) when \( B \) is first set to 0. In this case, the final state of the circuit is determined by the transient behavior, which depends on the order in which the state changes of the inputs take place. In general, if shifting from one state to another requires a change in at least two state variables, then a *race condition* will occur.
The race is said to be *non-critical* if the order in which the variables change state does not affect the final state of the circuit.

If, on the contrary, the circuit can assume two or more stable states depending on the order in which the variables change state, the race is said to be *critical*.

![Figures 1.2](image-url)

**Figure 1.2.** a) NAND gate based latch with initial conditions specified; b) logic circuit of the latch and representations useful for the determination of c) $X^+$ and d) $Y^+$

A NAND gate based latch with initial conditions specified is illustrated in Figure 1.2(a). Taking into account the fact that the differences in propagation delay of the two logic gates may translate into a delay, $\Delta$, between an output and the feedback input, an equivalence may be established between the latch in Figure 1.2(b) and each representation shown in Figures 1.2(c) and 1.2(d).

The following logic equations may be derived based on the circuit shown in Figure 1.2(c):

\[
X^+ = \overline{A} \cdot Y \quad [1.11]
\]

\[
Y = \overline{B} \cdot X \quad [1.12]
\]

By substituting [1.12] into [1.11], we obtain:

\[
X^+ = \overline{A} \cdot \overline{B} \cdot X \quad [1.13]
\]

\[
= \overline{A} + \overline{B} \cdot X \quad [1.14]
\]

\[
= A + \overline{B} \cdot X
\]
In the case of the circuit shown in Figure 1.2(d), the logic equations are written as follows:

\[ X = \overline{A} \cdot Y \]  \hspace{1cm} [1.15]

\[ Y^+ = \overline{B} \cdot X \]  \hspace{1cm} [1.16]

Substituting [1.15] into [1.16], we obtain:

\[ Y^+ = \overline{B} \cdot \overline{A} \cdot Y \]  \hspace{1cm} [1.17]

\[ = \overline{B} + \overline{A} \cdot Y \]  \hspace{1cm} [1.18]

The characteristic equations of the NAND gate based latch are therefore in the following form:

\[ X^+ = A + \overline{B} \cdot X \]  \hspace{1cm} [1.19]

and

\[ Y^+ = B + \overline{A} \cdot Y \]  \hspace{1cm} [1.20]

\[
\begin{array}{c|c|c|c}
\overline{A} & \overline{B} & X & X^+ & Y^+ \\
1 & 1 & 1 & 1 & 0 \\
1 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 1 \\
1 & 0 & 0 & 0 & 1 \\
0 & 1 & 1 & 1 & 0 \\
0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 \\
\end{array}
\]

Table 1.2. State table of the NAND gate based latch

The state table of the NAND gate based latch may be constructed, as shown in Table 1.2, based on characteristic equations and initial conditions.

We can see that the signals \( X^+ \) and \( Y^+ \) are complementary except when the two inputs \( \overline{A} \) and \( \overline{B} \) are set at 0.

In addition, the signals \( X^+ \) and \( Y^+ \) are only defined uniquely when the inputs \( \overline{A} \) and \( \overline{B} \) cannot change states from 0 to 1 simultaneously. Thus, the outputs of the latch
are defined by \((X, Y) = (0, 1)\) if the input \(\overline{A}\) is first set to 1 or by \((X, Y) = (1, 0)\) if the input \(\overline{B}\) is first set to 1. In this case, as the final state depends on the order in which the inputs change states, we have a critical race condition.

Among the combinations of states that the outputs of the latch can take, only those for which \(X^+ = X\) and \(Y^+ = Y\) are said to be \textit{stable}.

\subsection*{1.2.1. SR latch}

For the SR latch (S stands for \textit{set}, and R for \textit{reset}) represented in Figure 1.3, we can obtain the characteristic equations from equations [1.9] and [1.10], as follow:

\[Q^+ = \overline{R} \cdot S + \overline{R} \cdot Q = \overline{R} \cdot (S + Q)\]  
[1.21]

and:

\[\overline{Q}^+ = \overline{S} \cdot R + \overline{S} \cdot \overline{Q} = \overline{S} \cdot (R + \overline{Q})\]  
[1.22]

It must be noted that complementing \(Q^+\) does not yield \(\overline{Q}^+\). The state table is given in Table 1.3.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{sr_latch.png}
\caption{SR latch: a) logic circuit; b) symbol}
\end{figure}

\begin{table}[h]
\centering
\begin{tabular}{ccc|cc}
S & R & Q & \(Q^+\) & \(\overline{Q}^+\) \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 0 & 1 \\
0 & 1 & 1 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 \\
1 & 0 & 1 & 1 & 0 \\
1 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 \\
\end{tabular}
\caption{State table of the SR latch}
\end{table}
An SR latch whose initial condition is specified can also be characterized using the truth table shown in Table 1.4. The SR latch is said to be reset-dominant 0, as setting both inputs to 1 causes the output \( Q \) to change to 0.

![Timing diagram for the SR latch](image)

**Figure 1.4. Timing diagram for the SR latch**

Figure 1.4 shows the timing diagram of the SR latch where the different operating modes that appear in the truth table can be observed.

However, if the forbidden state \((S = R = 1)\) is considered as a do not care state, the state table takes the form given in Table 1.5. Constructing a Karnaugh map, as
shown in Figure 1.5, we obtain another version of the characteristic equation given by:

\[ Q^+ = S + Q \cdot \overline{R} \quad \text{and} \quad S \cdot R = 0 \]  \hspace{1cm} [1.23]

\[
\begin{array}{|c|c|c|c|}
\hline
\text{SR} & 00 & 01 & 11 & 10 \\
\hline
\text{Q} & 0 & 0 & 2 & x \\
\hline
\text{Q} & 1 & 1 & 0 & 1 \\
\hline
\end{array}
\]

\text{SR} & 00 & 01 & 11 & 10 \\
\hline
\text{S} & x & 6 & 1 & 4 \\
\hline
\text{R} & 0 & 1 & 3 & 2 \\
\hline
\end{array}

Figure 1.5. Karnaugh map for the SR latch. For a color version of this figure, see www.iste.co.uk/ndjountche/electronics2.zip

This last equation is used for applications where neither of the inputs \( S \) and \( R \) can take the state 1.

When a transition requires a change in state for at least two variables, an analysis based on Karnaugh maps, as shown in Figure 1.6, is necessary to detect the critical race conditions.

\[
\begin{array}{|c|c|c|c|c|}
\hline
\text{SR} & 00 & 01 & 11 & 10 \\
\hline
\text{Q} & 0 & 0 & 2 & 1 \\
\hline
\text{Q} & 1 & 1 & 0 & 1 \\
\hline
\end{array}
\]

\text{SR} & 00 & 01 & 11 & 10 \\
\hline
\text{S} & x & 6 & 1 & 4 \\
\hline
\text{R} & 0 & 1 & 3 & 2 \\
\hline
\end{array}

(a) \hspace{5cm} (b)

Figure 1.6. Karnaugh map: a) critical race; b) non-critical race. For a color version of this figure, see www.iste.co.uk/ndjountche/electronics2.zip
Let us consider that from the initial state, where $S = 1$, $R = 1$, and $Q = 0$, and which corresponds to the cell 6 in the Karnaugh map of Figure 1.6(a), both inputs $S$ and $R$ must be reset to zero.

The state of the input $S$ can change before that of the input $R$, or vice versa.

The arrows entered in the Karnaugh map are used to illustrate the response of the latch in each case.

In $SR$ terms, the transition $11 \rightarrow 01 \rightarrow 00$ is produced, and the output is maintained at the final state $Q^{+} = 0$, corresponding to cell 0, if the input $S$ changes first. However, if the input $R$ changes first, the transition will be $11 \rightarrow 10 \rightarrow 00$, and the final state of the output is then $Q^{+} = 1$, corresponding to cell 1.

In the case of Figure 1.6(b), the flip-flop is initially characterized by $S = 1$, $R = 0$ and $Q = 1$; this corresponds to the cell 5 in the Karnaugh map.

As a result of the possible transitions, $10 \rightarrow 00 \rightarrow 01$ when $S$ changes first, or $10 \rightarrow 11 \rightarrow 01$ when $R$ changes first, the output takes the same final state, $Q^{+} = 0$, corresponding to cells 3 or 2. This corresponds to a non-critical race condition.

We can verify that the only critical race condition in an SR latch occurs when the inputs $S$ and $R$ that are initially set to 1 are reset to 0.

### 1.2.2. $\overline{S \overline{R}}$ latch

An $\overline{S \overline{R}}$ latch can be implemented using NAND gates, as shown in Figure 1.7(a). Its symbol is represented in Figure 1.7(b). Based on the truth table shown in Table 1.6, we can note that the inputs are activated by low-level signals. The $\overline{S \overline{R}}$ latch is said to be *set-dominant 1*, as setting both inputs to 1 changes the output $Q$ to 1.

![Figure 1.7. $\overline{S \overline{R}}$ latch: a) logic circuit; b) symbol](image)

The effect of a race condition on the operation of the latch can be analyzed using a Karnaugh map.
By referring to Figure 1.8(a), we can see that the flip-flop is initially characterized by $\bar{S} = 0$ and $\bar{R} = 0$, and $Q = 1$ (cell 1). The transition of the inputs $\bar{S}$ and $\bar{R}$ to 1 involves a change in two state variables. If, due to the difference in propagation delays, the input $\bar{S}$ changes first, this translates to the transitions, $00 \rightarrow 10 \rightarrow 11$, and the final state of the output is $Q^+ = 0$ (cell 6). If, on the other hand, the input $\bar{R}$ changes first, the latch follows the transitions, $00 \rightarrow 01 \rightarrow 11$, and the output takes the final state $Q^+ = 1$ (cell 7). This is a critical race condition because the final state of the outputs depends on the order in which the variables change.

![Karnaugh map: a) critical race; b) non-critical race.](www.iste.co.uk/ndjountche/electronics2.zip)

An example of a non-critical race condition is illustrated by the Karnaugh map, as shown in Figure 1.8(b). Starting from the state $\bar{S} = 1$ and $\bar{R} = 0$, and $Q = 0$ (cell 4), the inputs $\bar{S}$ and $\bar{R}$ must be set to 0 and 1, respectively. The two possible transitions $10 \rightarrow 00 \rightarrow 01$ (input $\bar{S}$ changes first) and $10 \rightarrow 11 \rightarrow 01$ (input $\bar{R}$ changes first) lead to the same final state for the output, $Q^+ = 1$ (cell 3 or 2).

For the $\bar{S} \bar{R}$ latch, the only critical race condition occurs when both inputs $\bar{S}$ and $\bar{R}$ move from 0 to 1.

<table>
<thead>
<tr>
<th>$\bar{S}$</th>
<th>$\bar{R}$</th>
<th>$Q^+$</th>
<th>$\bar{Q}^+$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>Q</td>
<td>$\bar{Q}$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 1.6. Truth table of the $\bar{S} \bar{R}$ latch
1.2.3. **Application: switch debouncing**

Contact bounces of a push-button switch (see Figure 1.9) during its closing or opening can be eliminated using a $\overline{S} \overline{R}$ latch, as shown in Figure 1.10, where $V_{CC}$ represents the supply voltage and $R_P$ is the polarization resistor.

![Waveform illustrating switch contact bounces](image1)

**Figure 1.9. Waveform illustrating switch contact bounces**

![Debouncing switch](image2)

**Figure 1.10. Debouncing switch**

When $\overline{R} = 0$, the output $Q$ of the latch is set to 1 as soon as the signal $\overline{S}$ reaches the logic level 1 for the first time. Subsequent fluctuations at the input $\overline{S}$ no longer affect the state of $Q$. Similarly, when $\overline{S}$ is at 0, the output $Q$ is reset to 0 following the first transition attributing the logic level 1 to $\overline{R}$.

1.3. **Gated SR latch**

A gated or level-sensitive SR latch uses a control signal $C$ that can be a clock signal. The signal $C$ is used to enable (or inhibit) the latch at specific time intervals.
1.3.1. Implementation based on an SR latch

The gated SR latch in Figure 1.11(a) is made up of two AND gates and an SR latch. It is represented by the symbol shown in Figure 1.11(b). It can be characterized by equations of the form:

\[ X^+ = A \cdot \overline{B} + \overline{B} \cdot X \] \[ \text{[1.24]} \]

and

\[ Y^+ = \overline{A} \cdot B + \overline{A} \cdot Y \] \[ \text{[1.25]} \]

where:

\[ A = RC, \quad B = SC, \quad X = Q, \quad X^+ = Q^+, \quad Y = \overline{Q}, \]

and \[ Y^+ = \overline{Q^+} \] \[ \text{[1.26]} \]

![Figure 1.11. Gated SR latch based on an SR latch: a) logic circuit; b) symbol](image)

The characteristic equations are, thus, given by:

\[ Q^+ = S \cdot C \cdot (\overline{R} \cdot C) + (\overline{R} \cdot C) \cdot Q \]

\[ = \overline{R} \cdot S \cdot C + (\overline{R} + C) \cdot Q \] \[ \text{[1.27]} \]

and

\[ \overline{Q^+} = (\overline{S} \cdot C) \cdot R \cdot C + (S \cdot C) \cdot \overline{Q} \]

\[ = R \cdot \overline{S} \cdot C + (\overline{S} + C) \cdot \overline{Q} \] \[ \text{[1.28]} \]

- If \( C = 0 \), we have \( Q^+ = Q \) and \( \overline{Q^+} = \overline{Q} \).
- If \( C = 1 \), we have \( Q^+ = \overline{R} \cdot (S + Q) \) and \( \overline{Q^+} = \overline{S} \cdot (R + \overline{Q}) \).
Table 1.7 presents the state table of the gated SR latch based on an SR latch. The truth table can be constructed as shown in Table 1.8. An example of the timing diagram is illustrated in Figure 1.12, for the case where \( Q = 0 \) and \( \overline{Q} = 1 \) initially.

<table>
<thead>
<tr>
<th>C</th>
<th>S</th>
<th>R</th>
<th>Q</th>
<th>( Q^+ )</th>
<th>( \overline{Q^+} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>x</td>
<td>x</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>x</td>
<td>x</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>x</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>x</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>x</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Table 1.7. State table of the gated SR latch based on an SR latch**

<table>
<thead>
<tr>
<th>C</th>
<th>S</th>
<th>R</th>
<th>( Q^+ )</th>
<th>( \overline{Q^+} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>x</td>
<td>x</td>
<td>( Q )</td>
<td>( \overline{Q} )</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>( Q )</td>
<td>( \overline{Q} )</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Table 1.8. Truth table of the gated SR latch based on an SR latch**

Figure 1.12. *Timing diagram of the gated SR latch*
1.3.2. Implementation based on an $\overline{S} \overline{R}$ latch

Another version of the gated SR latch, whose logic circuit and symbol are given in Figures 1.13(a) and 1.13(b), is implemented using two NAND gates and an $\overline{S} \overline{R}$ latch. By performing its analysis, the following equations can be derived:

$$X^+ = A + \overline{B} \cdot X$$ \[1.29\]

and:

$$Y^+ = B + \overline{A} \cdot Y$$ \[1.30\]

where:

$$A = \overline{S} \cdot C, \quad B = R \cdot C, \quad X = Q, \quad X^+ = Q^+, \quad Y = Q, \quad \text{and} \quad Y^+ = Q^+$$ \[1.31\]

and finally we have:

$$Q^+ = S \cdot C + (\overline{R} + \overline{C}) \cdot Q$$ \[1.32\]

and:

$$\overline{Q^+} = R \cdot C + (\overline{S} + \overline{C}) \cdot \overline{Q}$$ \[1.33\]

Figure 1.13. Gated SR latch based on an $\overline{S} \overline{R}$ latch: a) logic circuit; b) symbol

The truth table of the gated SR latch based on an $\overline{S} \overline{R}$ latch can, therefore, be constructed as shown in Table 1.9.
Table 1.9. Truth table of the gated SR latch based on an $\overline{S}$ $\overline{R}$ latch

<table>
<thead>
<tr>
<th>$C$</th>
<th>$S$</th>
<th>$R$</th>
<th>$Q^+$</th>
<th>$\overline{Q}^+$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>x</td>
<td>x</td>
<td>$\overline{Q}$</td>
<td>$Q$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>$Q$</td>
<td>$\overline{Q}$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

No change
Reset
Set
Forbidden state

Figure 1.14. Gated D latch: a) and b) logic circuits; c) symbol

1.4. Gated D latch

A gated D latch (D stands for data) can be implemented from a gated SR latch, as shown in Figure 1.14. Connecting an inverter between the $S$ and $R$ inputs prevents the forbidden state from occurring. By inserting the expressions:

$$R = \overline{D} \quad \text{and} \quad S = D \quad \quad \quad [1.34]$$

in any of the following two characteristic equations of the gated SR latches:

$$Q^+ = \overline{R} \cdot S \cdot C + (\overline{R} + \overline{C}) \cdot Q \quad \quad \quad [1.35]$$

and

$$Q^+ = S \cdot C + (\overline{R} + \overline{C}) \cdot Q \quad \quad \quad [1.36]$$

we obtain, for the gated D latch, the same characteristic equation, given by:

$$Q^+ = D \cdot C + D \cdot Q + \overline{C} \cdot Q \quad \quad \quad [1.37]$$

$$= D \cdot C \cdot (Q + \overline{Q}) + D \cdot (C + \overline{C}) \cdot Q + (D + \overline{D}) \cdot \overline{C} \cdot Q$$

$$= D \cdot C \cdot (Q + \overline{Q}) + \overline{C} \cdot Q \cdot (D + \overline{D})$$

$$= D \cdot C + \overline{C} \cdot Q \quad \quad \quad [1.37]$$
– If $C = 1$, the characteristic equation becomes $Q^+ = D$.
– If $C = 0$, we have $Q^+ = Q$.

With a gated $D$ latch, the state of the input $D$ is transferred to the output when the control (or enable) input $C$ is set to 1, while the state of the output does not change when the control input is reset to 0; this translates into a characteristic equation of the form:

$$Q^+ = D \cdot C + \overline{C} \cdot Q$$  \[1.38\]

The gated $D$ latch is thus said to be transparent when $C = 1$. It is, therefore, sensitive to the high level of the signal applied at the input $C$.

Figure 1.15 shows the symbol of a gated D latch. The truth table of a gated D latch is represented in Table 1.10, where the outputs $Q^+$ and $Q^+$ are complementary. An example of the timing diagram for the D latch is given in Figure 1.16, where the output $Q$ is initially set to 0.

**Table 1.10. Truth table of the gated D latch**

<table>
<thead>
<tr>
<th>$C$</th>
<th>$D$</th>
<th>$Q^+$</th>
<th>$Q^+$</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>x</td>
<td>Q</td>
<td>$\overline{Q}$</td>
<td>No change</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>Reset</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Set</td>
</tr>
</tbody>
</table>

**Figure 1.15. Symbol of the gated D latch**

1.5. Basic JK flip-flop

The JK flip-flop (J as a set input, and K as a reset input) is the most versatile of the basic flip-flops. When it is activated, it permits the storage of a binary data based on the combination of states taken by the inputs J and K. A JK flip-flop can be implemented
by using the logic circuit given in Figure 1.17(a). It is symbolically represented as shown in Figure 1.17(b). From the logic circuit of the JK flip-flop, we can obtain:

\[
S = J \cdot C \cdot \overline{Q} \quad \text{and} \quad R = K \cdot C \cdot Q
\]  

\[1.39\]

\[\text{Figure 1.16. Timing diagram for the gated D latch}\]

\[\text{Figure 1.17. Basic JK flip-flop: a) logic circuit; b) symbol}\]

By inserting these last expressions in the characteristic equation of the gated SR latch:

\[
Q^+ = \overline{R} \cdot (S + Q)
\]  

\[1.40\]

we get

\[
Q^+ = (K \cdot C \cdot \overline{Q}) \cdot (J \cdot C \cdot \overline{Q} + Q)
\]

\[
= (K + \overline{C} + \overline{Q}) \cdot (J \cdot C + Q)
\]

\[
= J \cdot K \cdot C + J \cdot \overline{Q} \cdot C + K \cdot Q + Q \cdot \overline{C}
\]

\[
= (1 + J \cdot C) \cdot K \cdot Q + (1 + K) \cdot J \cdot \overline{Q} \cdot C + Q \cdot \overline{C}
\]

\[
= J \cdot Q \cdot C + (K + \overline{C}) \cdot Q
\]  

\[1.41\]

– if \(C = 1\), the characteristic equation takes the form \(Q^+ = J \cdot \overline{Q} + K \cdot Q\);
– if \( C = 0 \), we have \( Q^+ = Q \).

The state table of the basic JK flip-flop can be constructed as shown in Table 1.11. The forbidden state, inherent to the SR latch, is eliminated by adding two feedback pathways in order to ensure that the output will be set to 1 only if \( Q = 0 \) and reset to 0 only if \( Q = 1 \). Table 1.12 presents the truth table of the basic JK flip-flop, where the outputs \( Q^+ \) and \( \overline{Q}^+ \) are complementary.

<table>
<thead>
<tr>
<th>( C )</th>
<th>( J )</th>
<th>( K )</th>
<th>( Q )</th>
<th>( Q^+ )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>Q</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 1.11. State table for the JK flip-flop

<table>
<thead>
<tr>
<th>( C )</th>
<th>( J )</th>
<th>( K )</th>
<th>( Q^+ )</th>
<th>( \overline{Q}^+ )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>x</td>
<td>x</td>
<td>Q</td>
<td>( \overline{Q} )</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Q</td>
<td>( \overline{Q} )</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>( \overline{Q} )</td>
<td>Q</td>
</tr>
</tbody>
</table>

Table 1.12. Truth table of the basic JK flip-flop

It must be noted that this JK flip-flop structure may be affected by undesirable oscillations. In fact, when the two inputs \( J \) and \( K \) are set at 1 and the clock signal changes to 1, the feedback of the values \( Q \) and \( \overline{Q} \) taken by the outputs forces the flip-flop to toggle (or to switch from one state to its logical complement). And if the clock signal is still at the logic state 1, the process recommences and the flip-flop again changes state. To ensure smooth operation, the pulse width of the clock signal must be smaller than the propagation delay of the flip-flop.

### 1.6. T flip-flop

A JK flip-flop can be transformed into a \( T \) flip-flop (\( T \) stands for toggle), as shown in Figure 1.18. When the \( T \) flip-flop is activated, its outputs change state every time a
pulse is applied to the input $T$. The characteristic equation of the JK flip-flop is given by:

$$Q^+ = J \cdot \overline{Q} \cdot C + (\overline{K} + \overline{C}) \cdot Q$$  \[1.42\]

Figure 1.18. $T$ flip-flop: a) logic circuit; b) symbol

Assuming that $J = K = T$, we obtain the characteristic equation of the $T$ flip-flop:

$$Q^+ = T \cdot \overline{Q} \cdot C + (T + \overline{C}) \cdot Q$$  \[1.43\]

- if $C = 1$, the characteristic equation is reduced to $Q^+ = T \cdot \overline{Q} + T \cdot Q = T \oplus Q$;
- if $C = 0$, we have $Q^+ = Q$.

Table 1.13 shows the state table of the $T$ flip-flop. As the outputs $Q^+$ and $\overline{Q}^+$ are complementary, the truth table for the T flip-flop can be constructed as shown in Table 1.14.

<table>
<thead>
<tr>
<th>$C$</th>
<th>$T$</th>
<th>$Q$</th>
<th>$Q^+$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$x$</td>
<td>$x$</td>
<td>$Q$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 1.13. State table of the $T$ flip-flop

<table>
<thead>
<tr>
<th>$C$</th>
<th>$T$</th>
<th>$Q^+$</th>
<th>$\overline{Q}^+$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$x$</td>
<td>$Q$</td>
<td>$\overline{Q}$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>$Q$</td>
<td>$\overline{Q}$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>$\overline{Q}$</td>
<td>$Q$</td>
</tr>
</tbody>
</table>

Table 1.14. Truth table of the $T$ flip-flop
1.7. Master-slave and edge-triggered flip-flop

The operation of circuits implemented by coupling level-triggered flip-flops may become unpredictable, as the signal state can propagate from the output of one flip-flop to another as long as the clock signal is activated, thus preventing data storage.

One solution to this problem consists of using master-slave or edge-triggered flip-flops. This is implemented by memorizing only those state changes that occur on receiving one of the edges of the clock signal as illustrated in Figure 1.19.

![Figure 1.19. Clock signal (τ: pulse width; T: signal period)](image)

1.7.1. Master-slave flip-flop

A Master-slave type flip-flop is implemented by connecting two flip-flops, called master and slave, whose clock signals are complementary.

1.7.1.1. Master-slave D flip-flop

An edge-triggered D flip-flop can be implemented using a master-slave structure that is composed of two gated D latches (see Figures 1.20 and 1.22).

![Figure 1.20. Master-slave D flip-flop triggered by the clock signal rising edge: a) logic circuit; b) symbol](image)
The master latch (or first latch) remains sensitive to changes in the input logic state as long as it is activated by the clock signal, but the output of the slave latch (or the second latch) only changes at the edges of the clock signal, when the master latch becomes deactivated and its state can no longer change. Thus, the output of the master-slave flip-flop only reflects the input logic state when the clock signal goes from high to low or vice versa.

Figures 1.20(a) and 1.20(b) show the logic circuit and symbol, respectively, for a D flip-flop triggered by the rising edge of the clock signal (or positive-edge-triggered D flip-flop). Table 1.15 gives the truth table. Figure 1.21 shows the timing diagram for a D flip-flop triggered by the rising edge of the clock signal.

In the case of the D flip-flop triggered by the falling edge of the clock signal (or negative-edge-triggered D flip-flop), the logic circuit and symbol are as represented in Figures 1.22(a) and 1.22(b), respectively. The truth table is given in Table 1.16.
1.7.1.2. JK master-slave flip-flop

A JK master-slave flip-flop can be described using the logic circuit and symbol represented in Figures 1.23(a) and 1.23(b), respectively, while its operation is characterized by the truth table given in Table 1.17.
When the master flip-flop is activated, its output logic state is determined not only by the inputs \( J \) and \( K \), but also by the outputs, \( Q \) and \( \overline{Q} \), of the slave flip-flop. The master flip-flop state is then transferred to the slave flip-flop only when the clock signal transitions from high to low (falling edge).

Thus, to ensure the normal operation of the JK master-slave flip-flop, the logic state taken by each input, \( J \) and \( K \), must not change when the master flip-flop is activated (or the clock signal \( CK \) is set to 1). If this condition is not satisfied, the outputs of the JK master-slave flip-flop may be affected by the undesirable catching of a logic state 1 or 0 by the master flip-flop:

– when the output \( Q \) of the slave flip-flop is at 0, the transition from 0 to 1 of the input \( J \) when \( CK = 1 \) results in the master flip-flop output being set at 1, and the slave flip-flop output can then be set to 1 when \( CK \) goes from 1 to 0. Once the master flip-flop is set to 1 following a change to 1 in the input \( J \), a subsequent assignment of 1 to the input \( K \) when \( CK = 1 \) cannot bring the master flip-flop output back to 0. This is because the slave flip-flop remains in the same state until the clock signal, \( CK \), again changes to 0 and the feedback signal \( Q = 0 \) keeps the input \( K \) deactivated. This behavior is known as 1s catching;

– in the case where the slave flip-flop output is at 1 and a transient disturbance forces the input \( K \) to change to 1 while \( CK = 1 \), the master flip-flop acquires this reset condition, which is then transferred to the slave flip-flop when the clock signal \( CK \) goes from 1 to 0. It must be noted that \( K \) subsequently changing to 1 while \( CK = 1 \) has no effect on the master flip-flop that can only be set to 1 by a high-going pulse at the input \( J \), which is actually deactivated by the feedback signal \( \overline{Q} = 0 \). This phenomenon is called 0s catching.
The JK master-slave flip-flop can be considered to be level triggered. The symbol $\uparrow$ is used in Figure 1.23(b) to indicate that the outputs of the JK master-slave flip-flop only reflect the state of the $J$ and $K$ inputs at the end of the pulse of the clock signal $CK$.

The truth table (see Table 1.17) is constructed assuming that the input signals $J$ and $K$ remain constant while the clock signal is set to 1 and, thus, does not take into account the 1s catching and 0s catching problem.

Figure 1.24 shows a timing diagram showing a 1s catching and 0s catching in a JK master-slave flip-flop.

![Timing diagram for the JK master-slave flip-flop](www.iste.co.uk/ndjountche/electronics2.zip)

**Figure 1.24.** Timing diagram for the JK master-slave flip-flop (illustration of 1s and 0s catching). For a color version of this figure, see www.iste.co.uk/ndjountche/electronics2.zip

### 1.7.2. Edge-triggered flip-flop

An edge-triggered flip-flop is designed so as to ensure that the output can only change at the rising or falling edge of the clock signal and remains constant between two consecutive edges.

#### 1.7.2.1. Principle of edge detection

Even though the circuits shown in Figure 1.25 are not exactly the same as those found in integrated flip-flops, they clearly demonstrate the detection principle of the edge of a signal.

The propagation delay caused by an inverter is exploited to generate a signal with a very small width during any transition of the clock signal.
1.7.2.2. Edge-triggered D flip-flop

In an edge-triggered D flip-flop, the detection of the clock signal transition in a given direction can be carried out by making use of the fact that the change in state (set, reset) of an SR or \( \overline{S} \overline{R} \) latch occurs only when the logic states of both inputs change. Thus, the state acquired by a latch after a clock signal transition occurs at one of the inputs while the other input is set to 1 or reset to 0; it cannot change only because of subsequent changes in the logic state of the clock signal.

Flip-flops can be triggered by the rising edge or the falling edge of the clock signal.

**D flip-flop triggered by the clock signal rising edge**

A D flip-flop triggered by the rising edge of the clock signal can be implemented using SR latches, as illustrated in Figure 1.26(a). The input signal \( D \) and the clock signal \( CK \) are applied to the input stage that generates the signal required by the output stage to determine the outputs \( Q \) and \( \overline{Q} \). When the clock signal goes from 0 to 1, the state (0 or 1) of the input \( D \) is converted by the input stage in a \( (\overline{S} \overline{R}) = (10) \) or \( (01) \) combination that results in the output stage being reset to 0 or set to 1. For other states that can be taken by the clock signal, the combination \( (\overline{S} \overline{R}) = (11) \) is generated by the input stage regardless of the value on the \( D \) input. This forces the output stage to maintain its logic level unchanged.

A triangle is placed at the clock signal input, as shown by the symbol in Figure 1.26(b), to indicate that the flip-flop is active on the rising edge of the clock signal. Table 1.18 gives the truth table.

**Flip-flop triggered by the clock signal falling edge**

Similarly, a D flip-flop triggered by the falling edge of the clock signal can be implemented using SR latches, as shown in Figure 1.27(a). Depending on whether the \( D \) input state is 0 or 1 the input stage generates, in response to a falling edge of the clock signal \( CK \), the combination \( (SR) = (01) \) or \( (10) \) that sets the output stage to 1 or resets the output stage to 0. For the other states that can be taken by the clock signal, the combination \( (SR) = (11) \) is generated by the input stage regardless of the value on the \( D \) input. This forces the output stage to maintain its logic level unchanged.

**Figure 1.25. Principle for detecting a) the rising edge and b) falling edge**
signal, the input stage produces the combination \((SR) = (00)\) and the output stage holds its previous state.

![Diagram](image)

**Figure 1.26.** \(D\) flip-flop triggered by the clock signal rising edge: 
   a) logic circuit; b) symbol

<table>
<thead>
<tr>
<th>D</th>
<th>CK</th>
<th>(Q^+)</th>
<th>(\overline{Q}^+)</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>0</td>
<td>Q</td>
<td>(\overline{Q})</td>
</tr>
<tr>
<td>x</td>
<td>1</td>
<td>Q</td>
<td>(\overline{Q})</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

**Table 1.18.** Truth table of the flip-flop

Referring to the symbol shown in Figure 1.27(b), a circle is placed before the triangle at the clock signal input to indicate that the flip-flop is activated by the clock signal falling edge. The truth table is represented in Table 1.19.

**APPLICATION.**– Implementation of a JK and \(T\) flip-flops using a \(D\) flip-flop
Figure 1.27. \textit{D} flip-flop triggered by the clock signal falling edge: 
a) logic circuit; b) symbol

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|}
\hline
D & CK & Q$^+$ & Q$^+$ \\
\hline
x & 0 & Q & \overline{Q} \\
\hline
x & 1 & Q & \overline{Q} \\
\hline
0 & \overline{\overline{0}} & 0 & 1 \\
\hline
1 & \overline{\overline{1}} & 1 & 0 \\
\hline
\end{tabular}
\caption{Truth table of the flip-flop}
\end{table}

A JK flip-flop triggered by the rising edge of the clock signal (or positive-edge-triggered JK flip-flop), as shown in Figure 1.28, can be implemented by adding a combinational circuit to a D flip-flop. The characteristic equation takes the following form:

\[ Q^+ = D = J \cdot \overline{Q} + \overline{K} \cdot Q \]  \[1.44\]
where $Q$ is the present state and $Q^+$ represents the next state. Table 1.20 shows the truth table. The timing diagram of the JK flip-flop triggered by rising edge is given in Figure 1.29.

![JK flip-flop triggered by the clock signal rising edge: a) logic circuit; b) symbol](image)

**Figure 1.28.** JK flip-flop triggered by the clock signal rising edge: a) logic circuit; b) symbol

<table>
<thead>
<tr>
<th>J</th>
<th>K</th>
<th>CK</th>
<th>$Q^+$</th>
<th>$\overline{Q^+}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>x</td>
<td>0</td>
<td>$\overline{Q}$</td>
<td>$\overline{Q}$</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>1</td>
<td>$\overline{Q}$</td>
<td>$\overline{Q}$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td></td>
<td>$\overline{Q}$</td>
<td>$\overline{Q}$</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td></td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td></td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td></td>
<td>$\overline{Q}$</td>
<td>$Q$</td>
</tr>
</tbody>
</table>

Table 1.20. Truth table

We can also implement a $T$ flip-flop by connecting a combinational circuit to a D flip-flop as illustrated in Figure 1.30. If $Q$ is the present state and $Q^+$ denotes the next state, the characteristic equation for the edge-triggered flip-flop is given by:

$$Q^+ = D = T \oplus Q$$

[1.45]
Figure 1.29. Timing diagram of the $JK$ flip-flop triggered by the clock signal rising edge.

Figure 1.30. $T$ flip-flop triggered by the clock signal rising edge: a) logic circuit; b) symbol

Table 1.21. Truth table

Table 1.21 gives the corresponding truth table. The timing diagram of the $T$ flip-flop triggered by the clock signal rising edge is represented in Figure 1.29.
1.8. Flip-flops with asynchronous inputs

Just after power-up, for instance, asynchronous inputs can be used to define initial conditions of a flip-flop, regardless of the states of synchronous inputs and the clock signal in order to prevent any possible hazards. They are generally low active.

– The D flip-flop shown in Figure 1.32(a) has two asynchronous inputs, $\overline{PR}$ and $\overline{CLR}$, that can be used to determine the output state, regardless of the clock signal. Its symbol is given in Figure 1.32(b). Based on the truth table, shown in Table 1.22, the input $PR$ sets the output to 1 (asynchronous preset), and $CLR$ resets the output to 0 (asynchronous clear). For operation in the synchronous mode, the inputs $PR$ and $CLR$ must be kept in the high logic state.

– A JK flip-flop triggered by the clock signal falling edge (or negative edge triggered JK flip-flop), as depicted in Figure 1.34(a), consists of a synchronous SR
Latch and Flip-Flop

Latch connected to NAND gates. It can be set to 1 or reset to 0 using the asynchronous inputs $PR$ and $CLR$, respectively. Its symbol is shown in Figure 1.34(b).

<table>
<thead>
<tr>
<th>PR</th>
<th>CLR</th>
<th>D</th>
<th>CK</th>
<th>$Q^+$</th>
<th>$\overline{Q^+}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>x</td>
<td>x</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>x</td>
<td>x</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>x</td>
<td>x</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>x</td>
<td>0</td>
<td>$Q$</td>
<td>$\overline{Q}$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>x</td>
<td>1</td>
<td>$Q$</td>
<td>$\overline{Q}$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>$\nearrow$</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>$\nearrow$</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Asynchronous preset
Asynchronous clear
Forbidden state

Table 1.22. Truth table of the D flip-flop with asynchronous inputs

Normal operation

During a normal (or synchronous) operation of the flip-flop, the asynchronous inputs are set to 1. When the clock signal changes from 1 to 0, the 0 logic state is directly transferred to the output gated latch of the type SR, which is then activated so that the state of the inputs $J$ and $K$ can be taken into account. Because the NAND gates are sized to have a propagation delay in the order of the time required by the flip-flop outputs to change states, just enough time passed before the clock signal propagating through the NAND gates can affect the flip-flop, thereby preventing any other change in logic state. When the clock signal CK takes the 0 logic state, each NAND gate is then set to 1 and the state of the AND gate connected to the output is now only dependent on the feedback signal. This allows the flip-flop to preserve its earlier state. When the clock signal CK changes from 0 to 1, or takes the logic state 1,
the output of each AND gate directly connected to the clock signal is dependent only on the feedback signal. This prevents the flip-flop from changing state.

![Logic circuit and symbol of the JK flip-flop with asynchronous inputs](image)

Figure 1.34. Logic circuit and symbol of the JK flip-flop with asynchronous inputs

Edge triggering is implemented by exploiting the difference in propagation delays associated with the clock signal CK, that is applied directly and via the NAND gates to the SR latch.

Table 1.23 shows the truth table of a JK flip-flop with asynchronous inputs.

![Operational characteristics of an SR latch](image)

Figure 1.35. Operational characteristics of an SR latch

Note.— By simultaneously applying a data $D$ to the input $J$ and its complement, $\overline{D}$, to the input $K$, the JK flip-flop operates as a $D$ flip-flop.
1.9. Operational characteristics of flip-flops

A flip-flop only acquires a signal whose level can remain stable for a certain time. Thus, it can operate normally only when the setup time requirements are met.

The timing diagram shown in Figure 1.35 illustrates the effect of the following characteristics on the state of the $Q$ output of an SR latch:

- propagation delay $t_p$: this is the interval of time between the application of an input signal and the appearance of the resulting signal at the output. The delay $t_{pLH}$ is measured on the rising edge of the output, while $t_{pHL}$ is measured on the falling edge;

- minimum pulse width $\tau_m$: in order for the flip-flop to operate reliably, the width of each pulse must be greater than $\tau_m$, otherwise the state of the output may become metastable.

Flip-flops available in the form of integrated circuits have propagation delays of the order of a few nanoseconds.

In addition, with reference to waveforms of a synchronous D flip-flop shown in Figure 1.36, we can define:

- the *setup time* as the minimum time during which the input logic levels must be kept constant before the transition of the clock signal in order to ensure a reliable triggering;

<table>
<thead>
<tr>
<th>PR</th>
<th>CLR</th>
<th>CK</th>
<th>J</th>
<th>K</th>
<th>Q$^+$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td></td>
<td>0</td>
<td>0</td>
<td>Q</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td></td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td></td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td></td>
<td>1</td>
<td>1</td>
<td>$\overline{Q}$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>x</td>
<td>x</td>
<td>Q</td>
</tr>
</tbody>
</table>

*Table 1.23. Truth table of the JK flip-flop with asynchronous inputs*
– the hold time, which is the minimum time interval during which the logic levels of the input must be kept constant after the transition of the clock signal in order to guarantee a reliable triggering.

The set-up time and hold time for integrated-circuit flip-flops are of the order of a few nanoseconds. When the set-up and hold conditions are not satisfied, the output state of the flip-flop may become unpredictable (either 0 or 1). In some cases, we can observe an oscillation of the output signal or a metastable state situated between the high and low logic levels.

1.10. Exercises

EXERCISE 1.1.– Propose an equivalent switch-based circuit for each of the circuits in Figure 1.37.

What is the function of these circuits?

EXERCISE 1.2.– Consider the T latch whose logic circuit and symbol are given in Figure 1.38.

Determine the characteristic equations of this latch.

EXERCISE 1.3.– Analyze and construct the truth table for the flip-flop shown in Figure 1.39.

EXERCISE 1.4.– Consider the positive edge-triggered D flip-flop shown in Figure 1.40(a). Complete the timing-diagram in Figure 1.40(b).

EXERCISE 1.5.– Figure 1.41(a) shows a positive edge-triggered JK flip-flop. Complete the timing diagram in Figure 1.41(b).
EXERCISE 1.6.– Consider the master-slave JK flip-flop in Figure 1.42(a). Complete the timing diagram shown in Figure 1.42(b).

EXERCISE 1.7.– Figure 1.43(a) shows a JK flip-flop with asynchronous inputs. Complete the timing diagram in Figure 1.43(b).
EXERCISE 1.8.– The logic circuit for a D flip-flop with asynchronous inputs is represented in Figure 1.44(a). Complete the timing diagram in Figure 1.44(b).

EXERCISE 1.9.– For each circuit using two D flip-flops, as represented in Figures 1.45–1.47, complete the corresponding timing diagram.

EXERCISE 1.10.– Complete the timing diagram corresponding to each of the circuits using two JK flip-flops, as represented in Figures 1.48 and 1.49.

EXERCISE 1.11.– Consider the logic circuit shown in Figure 1.50(a), which is made up of two D flip-flops and a combinational logic circuit section F to be determined.
Figure 1.43. a) JK flip-flop; b) timing diagram

Figure 1.44. a) D flip-flop; b) timing diagram

Figure 1.45. a) Logic circuit 1; b) timing diagram

Complete the timing diagram (signals $Q_1$ and $Q_2$) in Figure 1.50(b).

Determine the logic function $F$ and suggest how it can be implemented.
EXERCISE 1.12.– Determine the characteristic equation for each of the synchronous D flip-flops in Figure 1.51. To compare these two D flip-flops, we use the set-up shown in Figure 1.52(a) and assume that the propagation delay of the inverter is not equal to zero.

Complete the timing diagram in Figure 1.52(b).

EXERCISE 1.13.– Converting between different types of flip-flops. Verify the equivalence between the flip-flops represented on each of the lines a, b, c and d in Figure 1.53.

1.11. Solutions

SOLUTION 1.1.– The equivalent circuit for each of the proposed circuits is represented in Figure 1.54.

It is a switch debouncer.

SOLUTION 1.2.– $T$ latch.

An SR latch is characterized by:

$$Q^+ = S \cdot C + (R + \overline{C}) \cdot Q$$  \[1.46\]
or

\[ Q^+ = R \cdot S \cdot C + (\overline{R} + \overline{C}) \cdot Q \]  \[1.47\]

Assuming that for the T latch, \( S = \overline{Q} \), \( R = Q \) and \( C = T \), we obtain the same characteristic equation in both cases, which can be written as follows:

\[ Q^+ = \overline{Q} \cdot T + \overline{T} \cdot Q = T \oplus Q \]  \[1.48\]
Figure 1.53. Flip-flops

Figure 1.54. Equivalent circuits
SOLUTION 1.3.– D flip-flop with enable input.

The characteristic equation for the D flip-flop with enable input is given by:

$$Q^+ = D \cdot EN + \overline{EN} \cdot Q$$

[1.49]

Figure 1.55 shows the logic circuit and the truth table for the D flip-flop with enable input is represented in Table 1.24.

![Figure 1.55. Logic circuit for D flip-flop with enable input](image)

<table>
<thead>
<tr>
<th>EN</th>
<th>D</th>
<th>CK</th>
<th>$Q^+$</th>
<th>$\overline{Q}^+$</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>x</td>
<td>0</td>
<td>Q</td>
<td>$\overline{Q}$</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>1</td>
<td>Q</td>
<td>$\overline{Q}$</td>
</tr>
<tr>
<td>0</td>
<td>x</td>
<td></td>
<td>Q</td>
<td>$\overline{Q}$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td></td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td></td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 1.24. Truth table of the flip-flop

SOLUTION 1.4.– Positive edge-triggered DD flip-flop.

For the positive edge-triggered D flip-flop, Figure 1.56 shows the logic circuit and the timing diagram that can be obtained from the truth table.

Figure 1.57 shows the logic circuit and the timing diagram for the level-triggered D flip-flop.
Figure 1.56. a) Positive edge-triggered D flip-flop; b) timing diagram

Figure 1.57. a) Level-triggered D flip-flop; b) timing diagram

Figure 1.58 shows the logic circuit and the timing diagram that can be used to compare these two types of D flip-flops.

Figure 1.58. Comparison of two D flip-flops: a) logic circuit; b) timing diagram

SOLUTION 1.5.– Positive edge-triggered JK flip-flop.

Figure 1.59 shows the logic circuit and timing diagram for the positive edge-triggered JK flip-flop.
Figure 1.59. a) JK flip-flop; b) timing diagram

SOLUTION 1.6.– Master-slave JK flip-flop.

Figure 1.60 shows the logic circuit and timing diagram for the master-slave JK flip-flop.

Figure 1.60. a) Master-slave JK flip-flop; b) timing diagram

SOLUTION 1.7.– JK flip-flop with asynchronous inputs.

Figure 1.61 shows the logic circuit and the timing diagram for the JK flip-flop with asynchronous inputs.

SOLUTION 1.8.– D flip-flop with asynchronous inputs.

Figure 1.62 shows the logic circuit and timing diagram for the D flip-flop with asynchronous inputs.

SOLUTION 1.9.– Connection of two D flip-flops.

Figure 1.63 shows logic circuit 1 and the corresponding timing diagram.

Logic circuit 2 and its timing diagram are represented in Figure 1.64.

Figure 1.65 shows logic circuit 3 and the corresponding timing diagram that can be obtained based on the truth table.
Figure 1.61. a) JK flip-flop; b) timing diagram

Figure 1.62. a) D flip-flop; b) timing diagram

Figure 1.63. a) Logic circuit 1; b) timing diagram

SOLUTION 1.10.– Connection of two JK flip-flops.

Figure 1.66 shows logic circuit 1 and the timing diagram that can be obtained based on the truth table.
Figure 1.64. a) Logic circuit 2; b) timing diagram

Figure 1.65. a) Logic circuit 3; b) timing diagram

Figure 1.66. a) Logic circuit 1; b) timing diagram

Figure 1.67 shows logic circuit 2 and the corresponding timing diagram.

SOLUTION 1.11.– Circuit using D flip-flops.

The truth table for the level-triggered D flip-flop and the truth table for the positive edge-triggered D flip-flop can be used to complete the timing diagram (for the outputs $Q_1$ and $Q_2$) for the circuit shown in Figure 1.68(a), as illustrated in Figure 1.68(b).
Considering $Q_1$ and $Q_2$ as the inputs and $X$ as the output, the truth table (see Table 1.25) obtained based on the timing diagram helps define the logic relationship that exists between $Q_1$, $Q_2$ and $X$.

Because the resulting logic equation is of the form, $X = Q_1 + Q_2$, the function $F$ can be implemented by an OR gate (see Table 1.25).

<table>
<thead>
<tr>
<th>$Q_1$</th>
<th>$Q_2$</th>
<th>$X$</th>
</tr>
</thead>
<tbody>
<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>
</tbody>
</table>

Table 1.25. Truth table (OR gate)
SOLUTION 1.12.– Gated D latches.

By analyzing each latch, we obtain a characteristic equation of the following form:

– latch L1:

$$Q^+ = D \cdot C + Q \cdot \overline{C}$$  \hspace{1cm} [1.50]

– latch L2:

$$Q^+ = D \cdot C + Q \cdot \overline{C} + D \cdot Q$$  \hspace{1cm} [1.51]

![Logic Diagram](image)

**Figure 1.69.** a) Logic circuit; b) timing diagram

Figure 1.69 shows the timing diagram that can be used to compare the latches L1 and L2.

The operation of the latch L1 is affected by the propagation delay of the inverter used to generate the signal $\overline{C}$. Hence, if $D = 1$ and $Q = 1$, we have:

– latch L1:

$$Q^+ = C + \overline{C}$$  \hspace{1cm} [1.52]

– latch L2:

$$Q^+ = 1 + C + \overline{C} = 1$$  \hspace{1cm} [1.53]

Adding the redundant term $D \cdot Q$ corresponding to cells 5 and 7 of the Karnaugh map (see Figure 1.70) is useful for the elimination of the aforementioned functional hazard in the case of the latch L2.
SOLUTION 1.13.– Conversion of one type of flip-flop to another.

– T flip-flop

For the circuit based on the D flip-flop, we get:

\[ Q^+ = D = T \oplus Q \] \[\text{[1.54]}\]

Considering the circuit based on the JK flip-flop, we have:

\[ Q^+ = D = J \cdot \overline{Q} + K \cdot Q = T \cdot \overline{Q} + \overline{T} \cdot Q = T \oplus Q \] \[\text{[1.55]}\]

In both cases, we have the characteristic equation for the T flip-flop.

– JK flip-flop

By analyzing the circuit based on the D flip-flop, we can write:

\[ Q^+ = D = J \cdot \overline{Q} + K \cdot Q \] \[\text{[1.56]}\]

this is the characteristic equation of the JK flip-flop.

– D flip-flop

For the circuit based on the JK flip-flop, we have:

\[ J = D, \quad K = D, \quad \text{and} \quad Q^+ = J \cdot \overline{Q} + K \cdot Q = D \] \[\text{[1.57]}\]

this is the characteristic equation of the D flip-flop.
The logic expression obtained for the circuit based on the D flip-flop is of the form:

\[ Q^+ = d = D \cdot EN + Q \cdot EN \]  
[1.58]

The equation associated with the circuit based on the JK flip-flop is given by:

\[ Q^+ = J \cdot \overline{Q} + K \cdot Q \]  
[1.59]

where \( J = D \cdot EN \) and \( K = D \cdot EN \). By applying Boolean algebra theorems, we can successively find that:

\[ Q^+ = D \cdot EN \cdot \overline{Q} + D \cdot EN \cdot Q \]  
[1.60]

\[ = D \cdot EN \cdot \overline{Q} + (D + EN)Q \]

\[ = D(EN + Q) + Q \cdot EN \]

\[ = D \cdot EN + D \cdot Q(EN + EN) + Q \cdot EN \]

\[ = D \cdot EN(1 + Q) + Q \cdot EN(1 + D) \]

\[ = D \cdot EN + Q \cdot EN \]  
[1.61]

In both cases, the characteristic equation obtained is that of a D flip-flop with enable input.