# 5 Storage Elements

## Student Group

| First Name | Surname | Matrikel Nr. |
|---|---|---|
| | | |
| | | |
| | | |

## Table of Contents

# 5. Storage Elements

In the previous chapter we have had a look onto memory devices, which store data even when no voltage is applied. This is great for longterm storage, like measurement data, pictures or music. The clock frequency of the storage element are often much lower than the internal frequency of the processor / controller. By this, the processor has to wait for the stored information due to high access time.

Therefore an controller-internal memory is advantageous. These are often called cache. Distinct storage elements have special properties, e.g. the written data changes the logic level of a pin ('foot') of the IC directly. We will now focus onto these controller-internal, fast memory, which consist of logic gates.

The name flipflop stemms from the fact, that the smallest logic circuit for storing data has to store binary values. Therefore, it has to show one of two stable states, and can flip into the other one by an external interaction.

## 5.1 Evolution of a Flipflop

### Flipflop as a Blackbox

In order to understand the wanted storage element, we will first look onto these element based on the IPO model (input–process–output).

The process of the storage element is to store two different states. This property can be implemented via two inverting gates which are interconnected in a feedback loop. The simple setup would be with NOT gates as shown in figure 1.

Fig. 1: Storing two different states

Of course this simple elements misses inputs and outputs! Therefore we have to look into these now.

The input of this element needs at least two inputs. Often the following two are used:

- Set input: once this input is high, a $1$ is stored. This input is marked as $S$.
- Reset input: once this input is low, a $0$ is stored. This input is marked as $R$.

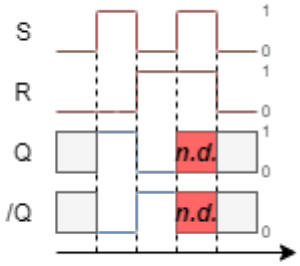
For the output also often two pins are shown. The pin $Q$ outputs the stored data directly. The pin $/Q$ outputs the inverted value.

Based on this simple requirements we can create the truth table.

- When $S=0$ and $R=0$, nothing changes and the outputs stay the same: $Q(n+1)=Q(n)$, $/Q(n+1)=/Q(n)$
- When $S=1$ and $R=0$, the stored information will be set: $Q(n+1)=1$, $/Q(n+1)=0$
- When $S=0$ and $R=1$, the stored information will be reset: $Q(n+1)=0$, $/Q(n+1)=1$
- When $S=1$ and $R=1$, it is unclear what to do.

Fig. 2: truth table for flipflop

| RS Flip Flop | | | |
|---|---|---|---|
| S | R | $Q_{(n+1)}$ | $/Q_{(n+1)}$ |
| 0 | 0 | $Q_{(n)}$ | $/Q_{(n)}$ |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | *n.d.* | *n.d.* |




In figure 2 the last input ($S=1$, $R=1$) reaches a not defined state. This state have to be investigated more later.

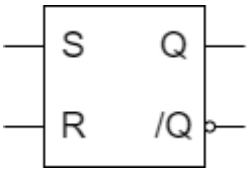
First, the storage device have get some inputs in order to change the stored stage. For This, a “switchable” NOT-gate is needed. Looking back to the chapter Boolean Algebra - Convertibility of Gates, this can be achieved by NAND gates or NOR gates.
In figure 3 a first approach is shown. But what how are the inputs $X_0$ and $X_1$ related to $S$ and $R$, as well as the outputs $Q$ and $/Q$ to $Y_0$ and $Y_1$?

Fig. 3: storage device based on NAND or NOR

In this introduction, only flip flops based on NOR gates are discussed - but flip flops are can also be build up with NAND gates. In figure 4 such a NOR flip flop is shown. Compared to figure 3 the outputs had to be rearranged in order to have the pins sorted as shown in the logic symbol (see figure 5)

Fig. 4: The RS flip flop (based on NOR gates

Fig. 5: The RS flipflop




### Flipflop as a Blackbox

# Examples

Fig. ##: Shift Register with synchronous Load

Fig. 31: Shift Register with synchronous Load, Direction Bit and Sout

# further Links

- https://www.electronics-tutorials.ws/sequential/seq_1.html
- https://www.electronics-tutorials.ws/counter/count_1.html