

6 Sequential Logic

Student Group

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

Table of Contents

- 6. Sequential Logic 2
- 6.1 State Diagram, State Transition Diagram 2
- 6.1.1 Motivation 2
- 6.1.2 Simple logic Example 3
- Moore Machine 6
- Mealy Machine 6
- Medvedev Machine 6

6. Sequential Logic

“I Know What You Did Last Cycle”

6.1 State Diagram, State Transition Diagram

6.1.1 Motivation

The diagrams of different states are well known from physics for example the state diagram (or better: phase diagram) of water, where its three states are: solid ice, liquid water and gaseous steam. The possible state transitions are due to temperature increase or decrease.

In [figure 9](#) image (1) the states of water are shown on the temperature axis. When only the state transitions are relevant, the states are simplified to a circle, showing the state name and behaviour. The transitions are depicted as arrows, where the needed condition is written onto (See [figure 9](#) image (2)). This diagram is called **state transition diagram**.



Fig. 9: States of Water

For matter not only the dimension “temperature” is important, but also the “pressure”. The full phase diagram is shown in [figure 10](#) image (1). By this, another variable is available and more transitions. These can be drawn into the state transition diagram ([figure 10](#) image (2)).



Fig. 10: States of Water

6.1.2 Simple logic Example

In German, often one has to pay for entering the toilet. An example of such an entrance control system is shown in figure 11. At this (artificial) example, one can pay either 50ct or 1€. Once paid, the turnstile will release and one can enter. Once the turnstile was pushed the entrance is closed again.



Fig. 11: Entrance Control for Toilets

The figure 12 the state transition diagram is drawn.

- The two states are that (1) the turnstile is opened and one is able to go through and (2) the turnstile is closed and one cannot enter anymore.
- The transitions are given by the done actions: one can either insert a coin or push on the turnstile.

Important here are some additional considerations:

- For the state transition diagram one has to **look for all possible transitions**. So, also pushing a closed turnstile or inserting more coins have to taken into account.
- A state transition diagram is not complete without a **legend** and without an **beginning/reset point**. The reset point is given by an arrow with "reset" written onto it

Fig. 12: State Transition Diagramm of the Entrance Control for Toilets



Out of this state transition diagram one can create a table-like representation, see figure 13.

Fig. 13: State Transition Diagramm of the Entrance Control for Toilets

| Toilet Entrance Control | | | |
|-------------------------|----------------|------------------|-------------------|
| current state | input / event | next state | output / action |
| turnstile closed | push turnstile | turnstile closed | disallow entrance |
| turnstile closed | insert coin | turnstile opened | allow entrance |
| turnstile opened | push turnstile | turnstile closed | disallow entrance |
| turnstile opened | insert coin | turnstile opened | allow entrance |

the inputs, outputs and states have to be encoded into binary, in order to investigate this table a bit more. How the binary value is connected to the outputs does not matter. We will choose the following coding:

- Encoding of the states: turnstile closed $\triangleq Q=0$, turnstile opened $\triangleq Q=1$,
- Encoding of the inputs: no coin inserted $\triangleq Xc=0$, coin inserted $\triangleq Xc=1$, turnstile not pushed $\triangleq Xp=0$, turnstile pushed $\triangleq Xp=1$,
- Encoding of the outputs: disallow entrance $\triangleq Y=0$, allow entrance $\triangleq Y=1$,

This table is shown in [figure 1](#) and is called **state transition table**.

Fig. 1: State Transition Diagramm of the Entrance Control for Toilets

| Toilet Entrance Control | | | | | | |
|-------------------------|--------|------|------|------------|------------|-----|
| current state | $Z(n)$ | Xc | Xp | next state | $Z(n + 1)$ | Y |
| | 0 | 0 | - | | 0 | 0 |
| | 0 | 1 | 0 | | 1 | 1 |
| | 1 | 0 | 1 | | 0 | 0 |
| | 1 | - | 0 | | 1 | 1 |



Interestingly, the logic circuit for this state transition table was already part of the course: it is the RS flip-flop! When looking deeper onto the table in [figure 1](#) one can substitute Xc with S (as in Set) and Xp with R (as in Reset) to directly get the truthtable of the RS flip flop.

Fig. ##: x

Moore Machine

- Number of flip-flops given by number of states
- synchronous output
- less logic gates necessary

Fig. 5: x

Mealy Machine

- Number of flip-flops given by number of states
- asynchronous output
- less logic gates necessary

Fig. 6: x

Medvedev Machine

- Number of flip-flops equal to number of outputs
- synchronous output
- more logic gates necessary

Fig. 7: x

From:

<https://mexle.te.hs-heilbronn.de/> - MEXLE Wiki

Permanent link:

https://mexle.te.hs-heilbronn.de/introduction_to_digital_systems/sequential_logic?rev=1637549990

Last update: **2021/11/22 03:59**

