



# Logische Zustände

## Übungen Digitales Design



### Lösung vs. Hinweise:

Nicht alle hier gegebenen Antworten sind vollständige Lösungen. Einige dienen lediglich als Hinweise, um Ihnen bei der eigenständigen Lösungsfindung zu helfen. In anderen Fällen wird nur ein Teil der Lösung präsentiert.

## 1 | LST - Logikgatter, welche nur einen Zustand liefern

### 1.1 Schalter-Schaltungen

$$y = \bar{a}$$

*lst/one-state-01-01*

### 1.2 Schalter-Schaltungen

$$y = \overline{ab}$$

*lst/one-state-01-02*

### 1.3 Schalter-Schaltungen

Multiple possible solution. Minimal with 2 interruptors n-type and 2 interruptors p-type.

*lst/one-state-01-03*



## 1.4 Open-Source Schaltung

Missing pull-down resistor on y.

<i>a</i>	<i>b</i>	<i>c</i>	<i>y</i>
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

*lst/one-state-02-01*

## 1.5 Alarmschaltung

Open-Drain and Open-Source Circuit possible.

*lst/one-state-02-02*

## 1.6 Zusammenstossaufspürung

You need either Open-Source or Open-Drain as well as a comparator. The priority depends on the Open-X you've chosen.

*lst/one-state-02-03*

## 1.7 Informationsübermittlung auf einem einzigen Draht

Possible with Open-Drain and Open-Source. Only one information can be transmitted at a time.

$$\begin{cases} D_a = '1' & \text{if } a = '0' \text{ and } b = '1' \\ D_b = '1' & \text{if } a = '1' \text{ and } b = '0' \end{cases} \quad (1)$$

*lst/one-state-02-04*



## 2 | LST - Logikgatter mit Ausgängen hoher-Impedanz

### 2.1 Serieverknüpfung von Peripheriebausteinen

Can be done with the help of a DEMUX-1to4.

*lst/hiz-01*

### 2.2 Erstellung einer Funktion Tristate Schaltungen

Two tri-state buffers are needed.

*lst/hiz-02*

### 2.3 Erstellung einer Funktion mit Hilfe von Tristate Schaltungen

You need one tri-state inverter.

*lst/hiz-03*

### 2.4 Zusammenstossaufspürung

You need a comparator and a buffer with tristate functionality. Only one component is enabled at any given time.

*lst/hiz-04*

### 2.5 Register mit bidirektionellem Datenbus

The register is written if **wr** = '1' and **cs** = '1' and the register is read if **rd** = '1' and **cs** = '1'. For the chip select get inspired by the exercises *lst\_hiz-01*.

*lst/hiz-05*