

2 Binäre Logik


Student Group

First Name	Surname	Matrikel Nr.

Table of Contents

<i>Motivation</i>	2
<i>Aufgaben</i>	2
Binäre Logik	2
<i>Ziele</i>	2
<i>Video</i>	2
Schaltzeichen der binären Logik	3
<i>Ziele</i>	3
<i>Video</i>	3
Anwendungen der binären Logik	3
<i>Ziele</i>	3
<i>Video</i>	3
weiterführende Links	4
<i>Anwendungen</i>	4

Motivation

Im ersten Kapitel haben wir  kennengelernt, wie die für Computer und Mikrocontroller relevanten Zahlensysteme zu verstehen sind. Damit lässt sich aber nicht verstehen, wie z.B. ein Programm abgearbeitet wird. Was wir dafür noch wissen müssen:

(Bild: [Sbp@Wikimedia](#), CC BY-SA 4.0)

Wie kann man mit den binären Zahlen logische Funktionen aufbauen?

Aufgaben

Die Aufgaben finden Sie in ILIAS unter: (2) Einführung, Vorlesungsskript und Hausarbeiten » Hausarbeiten

Details dazu finden Sie in der [Einführung zu Grundlagen der Digitaltechnik](#)

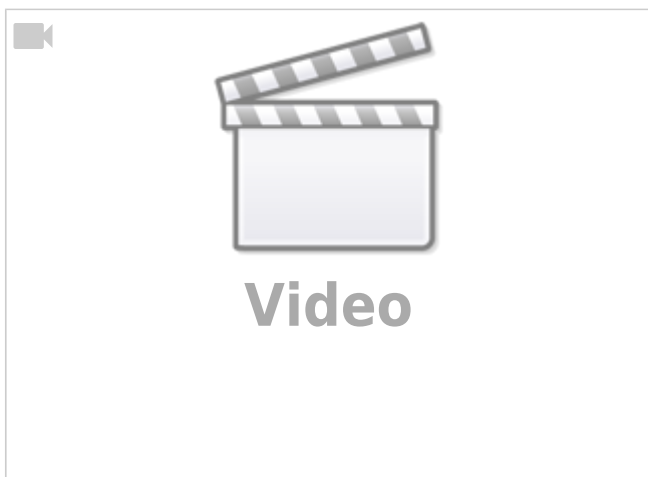
Binäre Logik

Ziele

Nach dieser Lektion sollten Sie:

1. die booleschen Funktionen, deren Schreibweisen und Funktionstabellen kennen.
2. die booleschen Rechenregeln anwenden können.
3. boolesche Ausdrücke vereinfachen können.
4. folgende Fachbegriffe kennen: (Logik)Gatter, Namen der Rechenregeln

Video



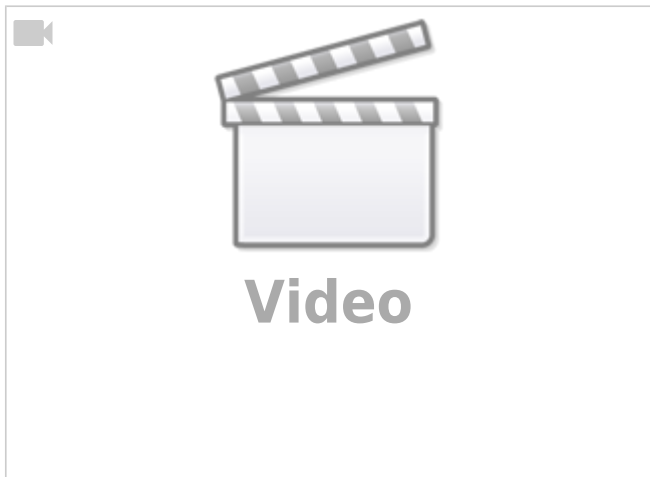
Schaltzeichen der binären Logik

Ziele

Nach dieser Lektion sollten Sie:

1. die beiden Arten der Schaltzeichen verstehen können.
2. Timing-Diagramme kennen.
3. Darstellung der Negierung von Ein-/Ausgängen kennen.
4. Gatter auf NAND und NOR-Gatter zurückführen können.

Video



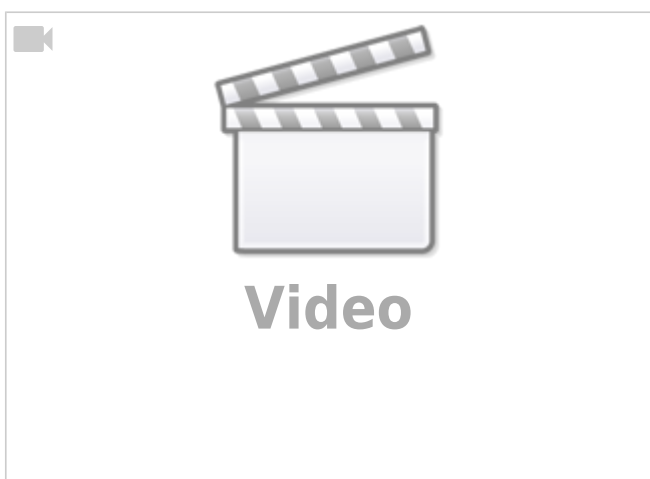
Anwendungen der binären Logik

Ziele

Nach dieser Lektion sollten Sie:

1. Zweck des Tri-State Gatters und des Zustands "Z" verstanden haben.
2. Die Anwendung des Zustands "Don't care" verstanden haben.
3. Zusammenschaltungen von wenigen Logikgattern in Funktionstabellen und umgekehrt wandeln können.
4. Gatter auf NAND und NOR-Gatter zurückführen können.

Video



weiterführende Links

- [Silicon Zoo](#): Hier ist die praktische Umsetzung von Logikgatter in Silizium zu sehen.

Anwendungen

- [Ethereum](#): Bei dieser Cryptowährung sind Rechnungen auf der Blockchain möglich. Dabei sind die logischen Grundfunktionen am günstigsten - ein Programm sollte also mit möglichst wenigen boole'schen Operatoren durchführbar sein
- [Beispiel in C](#) für die Verwendung der booleschen Algebra: durch Klick auf den Knopf Fork `this` ist der Code veränderbar.

From:

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

Permanent link:

https://mexle.te.hs-heilbronn.de/grundlagen_der_digitaltechnik/binaere_logik?rev=1602682426

Last update: **2021/05/09 09:59**

