Computerlogik 1+1=10

## Ziel des Projekts:

Dieses Projekt soll einen Einstieg in die Computerlogik geben. Wichtige Bestandteile dessen sind ein Grundverständnis für:

die Beschaffenheit von Halbleitern

Funktionsweise von Diode und Transistor

das binäre Zahlensystem

Logikgatter und dazugehörige Wahrheitstabellen

Des weiteren können praktische Experimente dazu genutzt werden obiges zu veranschaulichen und verständlicher zu machen.

## Diode und Transistor:

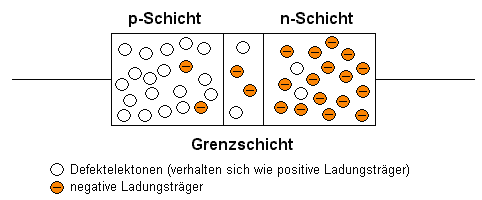


Abb. 1

Diode

Eine Diode besteht aus einem n- und einem p-dotierten Halbleiter, zwischen den beiden Schichten entsteht eine Sperrschicht (die Rekombinationszone). (siehe Abb.1).

Wird die Diode in Durchlassrichtung geschaltet, also den Pluspol an die p-Schicht und an die n-Schicht der Minuspol, muss zunächst eine bestimmte Schwellspannung überwunden werden, damit die Elektronen genug Energie haben, die Grenzschicht zu durchbrechen und damit ein Strom fließen kann. Die Elektronen bzw. die Elektronenlöcher werden vom gegenüberliegenden Pol angezogen und überbrücken auf diese Weise die Sperrschicht. (siehe Abb.2)

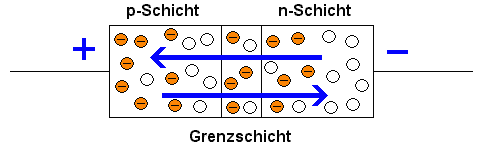


Abb.2

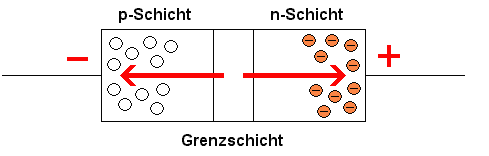
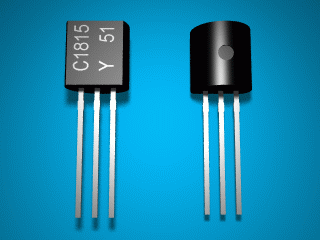


Abb.3

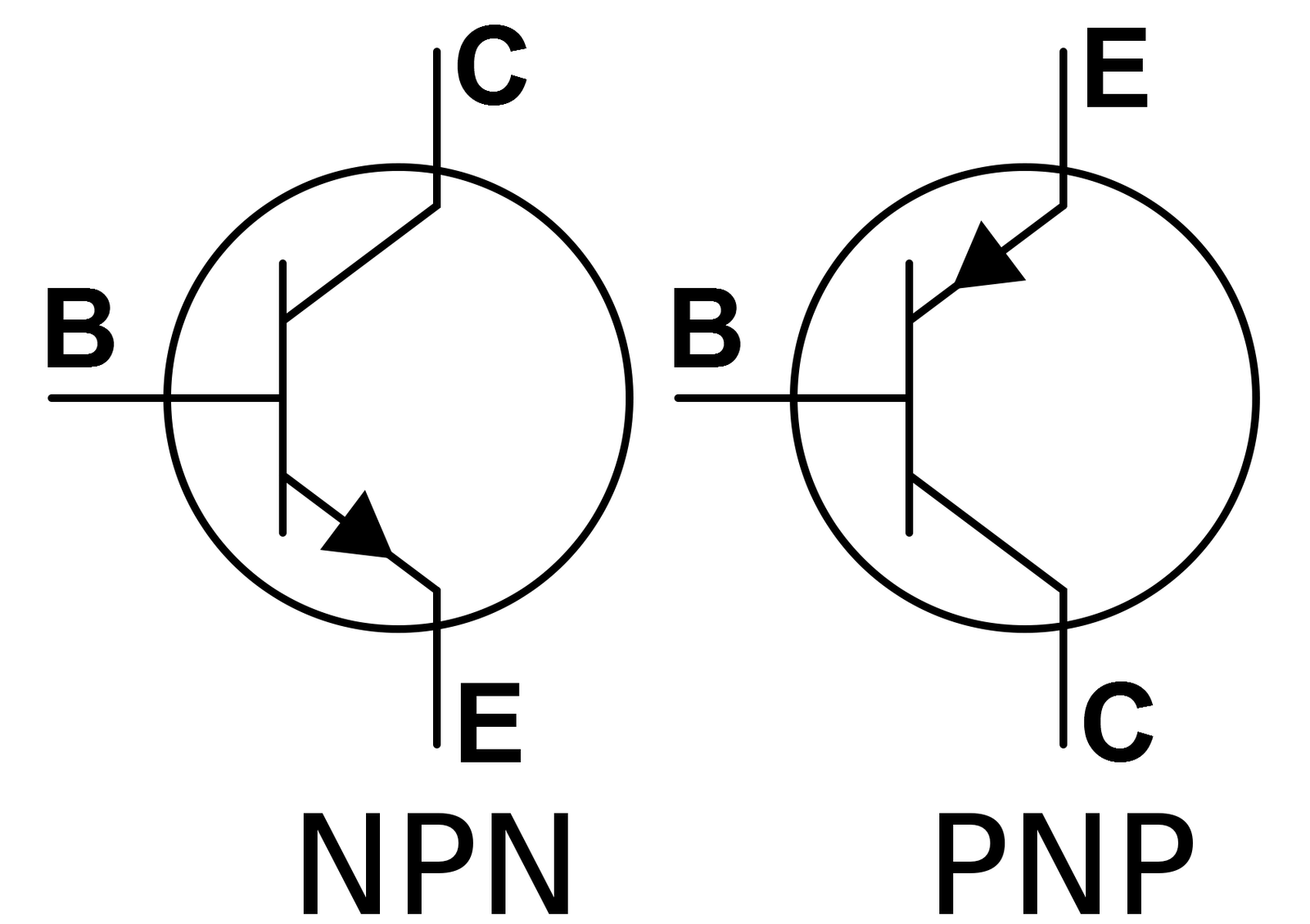
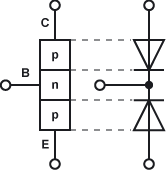
Wird die Diode jedoch andersrum geschaltet, ist die Diode in Sperrrichtung geschaltet. Das führt dazu, dass die Ladungsträge von den Polen angezogen werden, wodurch die Grenzschicht, also die ladungsfreie Zone größer wird, und kein Strom fließen kann. (siehe Abb.3)



Transistor

Ein Transistor funktioniert im grundlegenden wie ein Relais. Jedoch wird bei einem Transistor keine mechanische Bewegung benötigt um mit einem “kleinen” Stromkreis einen “großen” Stromkreis zu schalten.

Halbleiter machen es möglich, denn im Wesentlichen ist ein Transistor nichts anderes als zwei aneinander geschaltete Dioden. Ein Transistor besteht aus drei Schichten. Entweder sind die Schichtungen PNP oder NPN - wobei P eine positiv dotierte Schicht und N eine negativ dotierte Schicht darstellt.

Jede der drei Schichten hat einen elektronischen Kontakt. Sie heißen Basis(B), Kollektor(C) und Emitter(E).

Um Schaltzeichen besser identifizieren zu können hilft folgender Merksatz: “Tut der Pfeil der Basis weh (zeigt er auf sie), ist der Transistor PNP!”

Zwischen der Basis und dem Emitter fließt der “kleine Strom” (nur etwa 0,7V). Ist diese Durchbruchsspannung erreicht so werden erst du unteren beiden Schichten leitend und danach auch die obersten, somit kann von Kollektor zu Emitter der “große Strom” fließen.

In Kombination geschaltet lassen sich so bestimmte Ein- und Ausgabelogiken erstellen, die dann zur Steuerung von elektrischen Systemen benutzt werden können.

## Das binäre Zahlensystem:

Wie der Name schon erahnen lässt besteht das Zahlensystem aus zwei Ziffern (bi = zwei).

In der Regel geht man von zwei Ziffern (im späteren auch Zustände genannt) aus. Eins und Null.

Vorerst genügt zu wissen, dass Computer nur mit diesen beiden Zuständen umgehen können.

Um jedoch eine Verknüpfung zum Dezimalsystem zu haben (dezi = zehn), muss man sich mit der Umrechnung befassen.

Diese ist aus Zweierpotenzen zusammengesetzt und relativ einfach:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Binär | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
| Dezimal | 1 | 2 | 4 | 8 | 16 | 32 | 64 | 128 |

(Eine anschauliche Erklärung ist auch im Tafelwerk auf S. 10 zu finden!)

Hat man dieses System einmal verstanden, ist die Überleitung zum Computer recht naheliegend.

Ein Computer arbeitet mit elektrischen Signalen. Dabei kennt er nur “Strom” (“true”) und “kein Strom” (“false”). “True” wird der Einfachheit halber mit 1 dargestellt, “False” mit 0.

Die kleinste Einheit mit der der Computer rechnet ist ein Bit. Dieses entspricht genau einem Zustand - true oder false. Gehen wir von einem Byte, immer bestehend aus 8 Bit, aus, so lassen sich 512 verschiedene Zustände darstellen:

Von 00000000 = 0

Bis 11111111 = 255 (die Null mit eingerechnet ergeben sich so 256 Zustände)

Um den Aufbau eines Bits zu verdeutlichen nehmen wir die Dezimalzahl 149:

Binär ist das 10010101.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Binärziffer | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| Binär | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
| Dezimal | 1 |  | 4 |  | 16 |  |  | 128 |

WICHTIG!: Die letzte Ziffer des Bits ist die kleine 2er-Potenz (somit ist die erste Ziffer immer die größte 2er-Potenz)!

Um jetzt auf das gewünschte Ergebnis von 149 zu kommen muss man nur die mit “true” beantworteten Zustände addieren und man erhält die Dezimalzahl (1+4+16+128 = 149)

Natürlich kann eine Binärziffer aus mehr als nur acht Ziffern bestehen, das Prinzip bleibt dabei jedoch das Gleiche.

## Logikgatter und Wahrheitstabellen:

Damit Computer Entscheidungen treffen können, sind auch sie gewissen logischen Abläufen unterstellt. Diese werden Logikgatter genannt.

Normalerweise werden sie tabellarisch dargestellt, wobei man von zwei Eingaben und einer Ausgabe ausgeht. Diese Tabellen nennt man *Wahrheitstabellen.* Es werden alle möglichen Eingabekombinationen aufgelistet und die jeweilige Ausgabe zugeordnet.

|  | OR |  |
| --- | --- | --- |
| E1 | E2 | A |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Die einfacheren Gatter sind z.B.:

|  | XOR |  |
| --- | --- | --- |
| E1 | E2 | A |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

|  | XNOR |  |
| --- | --- | --- |
| E1 | E2 | A |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

|  | NOR |  |
| --- | --- | --- |
| E1 | E2 | A |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

|  | NAND |  |
| --- | --- | --- |
| E1 | E2 | A |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

|  | AND |  |
| --- | --- | --- |
| E1 | E2 | A |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Diese Gatter machen es möglich eine Logische Schaltung zu verstehen und gezielt zu verwenden.

Diese Schaltungen gibt es auch noch in “exklusiver” Form. Sie heißen dann *Exclusive OR, usw.* Außerdem kann man sie in negativer Form vorfinden. Sie bekommen dann die Vorsilbe “negated”, also *Negated AND, Negated OR, usw.*

Exclusive = X Negated = N

Natürlich kann man auch beide kombinieren: