

# Digitalelektronik



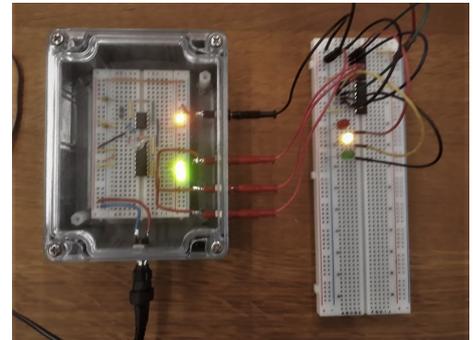
Die digitale Elektronik mit ihrer binären Schaltlogik ist die Grundlage der modernen Datenverarbeitung. In diesem Versuch werden elementare logische Verknüpfungen behandelt und einfache logische Schaltungen aufgebaut. Ein Teil der Aufgaben wird dabei mit Hilfe der Simulationssoftware LogicCircuit bearbeitet. Mit diesem Programm wird eine Ampelschaltung konstruiert, die anschließend auf einem Experimentierboard unter Verwendung handelsüblicher Logic-ICs aufgebaut wird. Schließlich wird die Funktion eines einfachen Inverters anhand einer Transistorschaltung demonstriert.

## 1 Lernziele

- Zahlensysteme (dezimal, binär, hexadezimal,...),
- elementare Boole'sche Algebra,
- einfache Schaltkreise (Halbaddierer, Volladdierer, RS-Flipflop),
- Ampelschaltung
- einfache Transistorschaltung

## 2 Experimenteller Aufbau

- Simulationsprogramm LogicCircuit,
- Experimentierboard,
- ICs: 6fach Inverter, 4fach AND-gate,
- NPN-Transistor BC547B
- Widerstände 100  $\Omega$ , 220  $\Omega$  und 1 k $\Omega$
- Multimeter zur Identifizierung der Widerstände.
- Steckverbinder, LED
- Steckernetzteil,
- 3-Kanal Binärzähler mit Netzteil



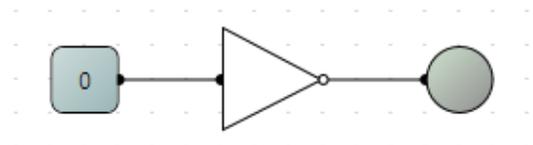
## 3 | Messung - | Durchführung - | Auswertung

### 3.1 Simulation digitaler Schaltungen - LogicCircuit

Die digitalen Schaltungen werden zunächst mit Hilfe des Programms LogicCircuit simuliert. Bitte beachten Sie, dass die graphische Darstellung der Logikelemente von der Spracheinstellung abhängt (siehe auch Abschnitt 5.1).

Starten Sie das Programm vom Desktop. Das Programmfenster, das sich nun öffnet, ist zweigeteilt mit den Bereichen „Project“ (links) und „Main“ (rechts). Die verschiedenen Elemente im linken Feld (möglicherweise müssen Sie die Listen „Input-Output“ oder „Primitives“ erweitern) können durch *drag and drop* in dem Programmfenster abgelegt werden.

Positionieren Sie auf diese Weise eine „1bit constant“, ein „NOT“-Gatter und eine LED nebeneinander in dem Programmfenster. Verbinden Sie mit gedrückter Maustaste die beiden schwarzen Anschlusspunkte der Konstanten und des NOT-Gatters und ebenso den offenen (negierten) Ausgang des NOT-Gatters mit dem Eingangsknoten der LED.



Aktivieren Sie das Programm durch Anklicken des Startknopfes  in der rechten unteren Ecke. Durch Doppelklick auf das Element der „1bit constant“ wechselt der Wert zwischen „0“ und „1“, das jeweils negierte Ergebnis wird durch die LED angezeigt. Stoppen Sie die Ausführung des Programms. Nehmen Sie sich nun ruhig ein paar Minuten Zeit, um andere logische Schaltelemente auszuprobieren. Starten Sie dann ein neues Projekt, *>File>New*.

### 3.2 Aufgabe 1: Halbaddierer (Simulation)

Unter einem Halbaddierer versteht man eine digitale Schaltung, mit deren Hilfe sich zwei einstellige Binärzahlen addieren lassen. Sie verfügt über zwei Ein- sowie zwei Ausgänge. An den beiden Ausgängen liefert der Halbaddierer einmal die direkte Summe der Zahlen, zum anderen den möglichen Übertrag.

- 1) Erstellen Sie in Ihrem Laborbuch eine vierspaltige Wertetabelle (zwei Eingänge und zwei Ausgänge) und tragen sie in die Eingangsspalten die vier möglichen Zustandskombinationen ein. Nun überprüfen Sie, welche Werte sich für die beiden Ausgangssignale ergeben, wenn Sie die Eingangssignale addieren. Nennen Sie einen Ausgang „Summe“ und den anderen „Übertrag“.
- 2) Realisieren Sie die Schaltung in der Simulation und testen Sie das Schaltverhalten. Bitte beachten Sie, dass eine Verbindung zwischen zwei Leitungen, z.B. an einer Abzweigung, immer durch einen schwarzen Punkt gekennzeichnet ist. Wenn dieser Punkt nicht automatisch gebildet wird, wählen Sie zuerst durch Linksklick das durchgehende Leitungselement in der Nähe der Verbindung und klicken Sie dann bei gedrückter ALT-Taste auf die Verbindungsstelle.
- 3) Der Halbaddierer wird in der folgenden Aufgabe als Schaltungselement weiterverwendet. Dazu wird die Schaltung als eigenständiger logischer Schaltkreis gespeichert. Markieren Sie die gesamte Schaltung und kopieren Sie sie in die Zwischenablage. Legen Sie nun ein leeres Schaltungselement an, `>Circuit>New logical circuit`. Doppelklick an einer leeren Stelle im Hauptfenster öffnet die Konfiguration des circuit-Elements. Tragen Sie einen Namen (z.B. Halbaddierer) und eine Notation (z.B. HA) ein. Beenden Sie die Konfiguration und fügen Sie die Schaltung aus der Zwischenablage ein. Ihr neu geschaffener Schaltkreis benötigt Anschlüsse für die Außenwelt. Ersetzen Sie dazu die 1bit-Konstanten durch Eingangspins („In-put 1bit pin“) und die LEDs durch Ausgangspins („Output 1bit pin“). Nach Doppelklick auf jedes Pin-Element können Sie die Namen der Anschlüsse ändern, (Eingänge A und B, Ausgänge S und Ü).
- 4) Wählen Sie nun unter „Circuit Project“ das Hauptfenster (Doppelklick) und löschen Sie die Elemente. Nun können Sie mit der Maus Ihren Halbaddierer in das Hauptfenster ziehen. Wenn Sie möchten, können Sie wieder Konstanten und LEDs anschließen, um die Funktion zu testen.

### 3.3 Aufgabe 2: Volladdierer (Simulation)

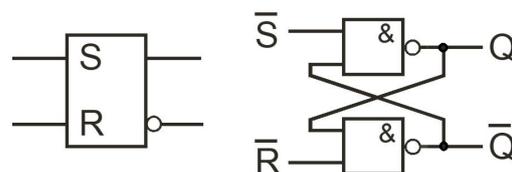
Bei der Addition mehrstelliger Binärzahlen muss neben der bitweisen Addition gleichwertiger Binärstellen gegebenenfalls auch der Übertrag aus der Addition der vorhergehenden geringerenwertigen Stelle berücksichtigt werden. Eine solche Schaltung heißt Volladdierer.

- 1) Erstellen Sie eine fünfspaltige Wertetabelle (drei Eingänge und zwei Ausgänge) und tragen Sie in die Eingangsspalten die möglichen Zustandskombinationen ein. Wie viele verschiedene Kombinationen gibt es? Überprüfen Sie auch hier, welche Ausgangswerte sich bei „Summe“ und „Übertrag“ durch die Addition der drei Eingangssignale ergeben.
- 2) Bauen Sie die Schaltung in LogicCircuit auf. Verwenden Sie dazu den Halbaddierer aus der ersten Aufgabe. Testen Sie die Schaltung und speichern Sie Ihr Projekt.

### 3.4 Aufgabe 3: RS-Flipflop (Simulation)

Ein Flipflop, auch bistabile Kippstufe genannt, ist eine elektronische Schaltung, die sich in zwei stabilen Zuständen befinden kann und eine „Speicherfunktion“ von 1 Bit besitzt. Es gibt eine große Vielzahl unterschiedlicher Flipflops, so dass man nur über diese Bauteile einen eigenen Praktikumsversuch konzipieren könnte. Hier soll nur das einfachste Flipflop, das RS-Flipflop, besprochen werden. Die Bezeichnung steht für die Funktionen *Set* und *Reset*.

Die Abbildung rechts zeigt das Schaltsymbol eines RS-Flipflops und dessen Aufbau durch zwei NAND-Gatter. Liegt am S-Eingang ein 1-Signal, springt der Ausgang von 0 auf 1. Dieser Zustand hält so lange an, bis am R-Eingang ein 1-Signal liegt. Beachten Sie, dass bei der Schaltung aus NAND-Gattern (rechts) die Eingänge negiert sind.



- 1) Erstellen Sie die Schaltung für ein RS-Flipflop aus zwei NAND-Gattern in LogicCircuit
- 2) Erstellen Sie Wertetabellen mit zwei Ein- und zwei Ausgängen und geben Sie die in der folgenden Tabelle vorgegebenen Eingangskombinationen der Reihe nach ein. Beobachten Sie, in welchem Zustand sich jeweils die beiden Ausgänge befinden und notieren Sie die Ergebnisse.

S	R	Q	$\bar{Q}$
0	0		
0	1		
0	0		
1	0		
0	0		

S	R	Q	$\bar{Q}$
1	0		
0	0		
0	1		
0	0		
0	1		

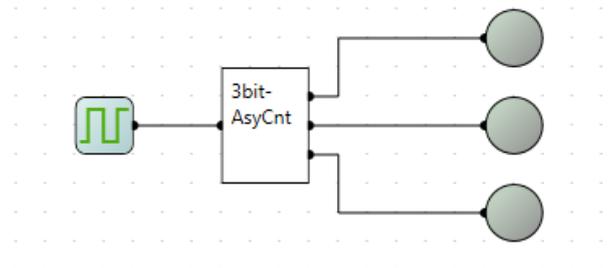
- a) Was fällt Ihnen auf und wie erklären Sie dieses Ergebnis?
- b) Was passiert, wenn an beiden Eingängen ein 1-Signal anliegt?

### 3.5 Aufgabe 4: Ampelschaltung (Simulation)

Entwerfen Sie den Schaltplan für die Signal-  
folge einer einzelnen, gewöhnlichen Verkehrs-  
ampel. Ein 3bit-Zähler erzeugt acht verschie-  
dene Zustände, von 0 (= binär 000) bis 7  
(=111), die den vier Ampelphasen, wie in der  
nebenstehenden Tabelle beschrieben, zuge-  
ordnet werden. Beim gleichmäßig getakteten  
Durchlaufen der Zustände werden die grüne  
und rote Phase dreimal länger angezeigt als  
die beiden Gelbphasen.

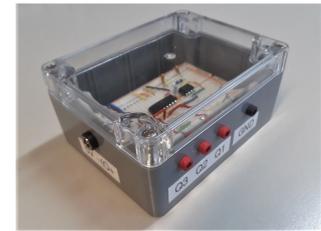
Zustand	Ampelphase	Eingang A	Eingang B	Eingang C
0	Rot			
1	Rot			
2	Rot			
3	Rot und Gelb			
4	Grün			
5	Grün			
6	Grün			
7	Gelb			

- 1) Füllen Sie zuerst die Tabelle aus. Übersetzen Sie dazu die Dezimalzahl der jeweiligen Zustände in die entsprechende Binärzahl, wobei jeder Eingang A-C für ein bit steht. Suchen Sie dann Gemeinsamkeiten in den bit-Mustern, die den gleichen Zustand erzeugen sollen. Die Ampel zeigt z.B. immer dann Rot an, wenn ...? Formulieren Sie diese Gemeinsamkeiten mit Hilfe der Booleschen Algebra und prüfen Sie, ob die Ausdrücke vereinfacht werden können. Es kann von Nutzen sein, nicht mit den Eingängen A, B und C, sondern mit den invertierten Eingängen zu arbeiten.
- 2) Für die Simulation der Ampelschaltung werden Sie einen bereitgestellten Binärzähler verwenden. Speichern Sie Ihr Projekt und öffnen (>File>Open) Sie das Projekt „Desktop/Digital-Elektronik/Grundpraktikum.CircuitProject“. Ziehen Sie den 3bit asynchronen Binärzähler auf das Programmierfeld Main und ergänzen Sie die Schaltung durch einen Taktgeber (Clock) sowie drei LEDs.
- 3) Starten Sie das Programm, verfolgen Sie das Zählen von 0 bis 7 im Binärformat und identifizieren Sie die Anschlüsse für die jeweiligen Zweierpotenzen.
- 4) Ergänzen Sie nun die Schaltung durch die Logikbauteile, mit denen Sie die Ampelsteuerung realisieren möchten. Verwenden Sie dabei die drei LEDs für die Ampel. Leider kann LogicCircuit LEDs nur in roter Farbe darstellen. Schreiben Sie daher die tatsächliche Ampelfarbe als Text neben die jeweilige LED.



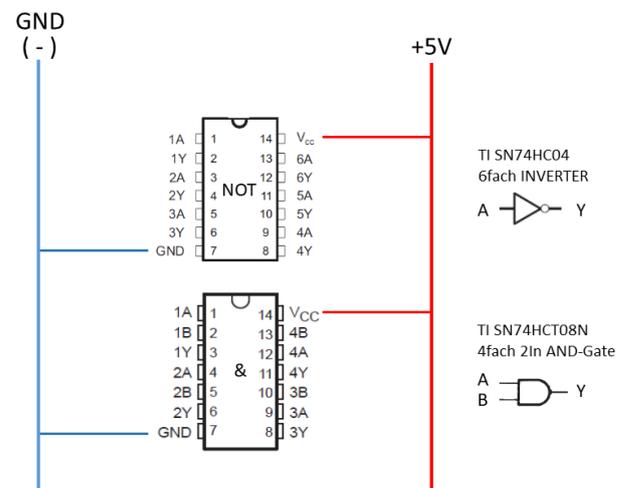
### 3.6 Aufgabe 5: Ampelschaltung (Experimentierboard)

Die nächste Aufgabe besteht darin, die von Ihnen entworfene Ampelschaltung mit elektronischen Bauelementen auf einer Experimentierplatine zu realisieren. Ein 3bit-Binärzähler (siehe Foto) wird zur Verfügung gestellt. An der Box finden Sie die Anschlüsse GND (Erdung, (-)-Pol), die drei Signalanschlüsse Q1 – Q3 und den Anschluss für ein 5V-Steckernetzteil.



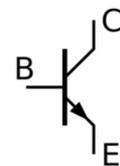
Auf der Experimentierplatine befinden sich zwei ICs, ein 6fach NOT-Gatter sowie ein 4fach AND (je zwei Eingänge). Verwenden Sie für die 5 Volt Spannungsversorgung der beiden ICs das Steckernetzteil am Arbeitsplatz (schwarzer Stecker = GND und roter Stecker = +5V).

- 1) Verdrahten Sie die Schaltung. Beachten Sie dabei folgende Hinweise
  - Auf der Experimentierplatine sind die Anschlusslöcher entlang der blauen und der roten Linien miteinander verbunden. Diese Anschlüsse dienen der Spannungsversorgung.
  - Innerhalb der Experimentierplatine sind jeweils links und rechts der Einkerbung die Anschlüsse mit derselben Nummer untereinander verbunden.
  - Prüfen Sie die Funktion der Leuchtdioden (kurzer Anschluss = Erde (-) und längerer Anschlussdraht = Signal (+)). Verwenden Sie für diesen Test und auch später in der Ampelschaltung  $100\ \Omega$  Vorwiderstände (in Serie zur LED) um den Strom zu begrenzen.
  - Beachten Sie, dass alle Schaltkreise (Experimentierplatine und Binärzähler) eine gemeinsame Erde (= GND = (-)-Pol) haben müssen.



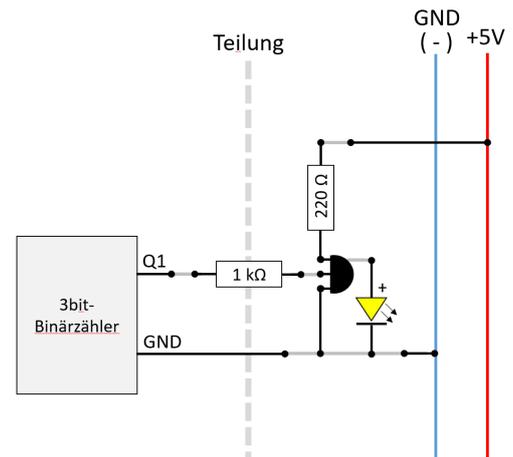
### 3.7 Aufgabe 6: Inverter-Schaltung mit einem NPN-Transistor (Experimentierboard)

Der erste funktionsfähige Computer Z3 wurde 1941 von Konrad Zuse und Helmut Schreyer gebaut und basierte auf mechanischen Relais. Die Entwicklung der Digitaltechnik wäre jedoch ohne die Erfindung des Transistors nicht möglich gewesen. Am Ende dieses Versuchs sollen Sie daher eine elementare Transistorschaltung mit einer NOT-Funktion kennenlernen. Bauen Sie auf der Experimentierplatine die in Kapitel 5.2, Abbildung 1 gezeigte Schaltung auf. Die Abbildung rechts zeigt das Schaltsymbol sowie die Pin-Belegung des Transistors BC547B.



Standard Pinning  
1 = C 2 = B 3 = E

- 1) Versuchen Sie bitte, die Bauteile möglichst wenig zu verformen, damit sie mehrfach verwendet werden können.
- 2) Verwenden Sie einen  $1\text{ k}\Omega$  Vorwiderstand an der Basis und einen  $220\ \Omega$  Widerstand am Kollektor.
- 3) Schließen Sie ein Netzteil an die durchgehend kontaktierte Reihe entlang der blauen (Erde) und roten Linie (5V) an.
- 4) Mit einer LED (Pluspol = langer Anschlussdraht) wird der logische Wert am Ausgang X (Kollektor) angezeigt.



- a) Wie hängt dieser Pegel X vom Eingangswert 0 (0V) oder 1 (+5V) an der Basis ab?
- b) Erläutern Sie die Funktion anhand des variablen Transistorwiderstandes im Spannungsteiler.

## 4 Vorbereitung, Fragen und Berechnungen vor Versuchsantritt

### 4.1 Vorbereitung

Informieren Sie sich zur Vorbereitung auf das Praktikum über die Themen, die oben als Lernziele genannt sind. Lesen Sie sorgfältig Kapitel 5. Wir empfehlen außerdem die Themen „Digitalschaltungen“ und „Transistoren“ in den unten aufgeführten Lehrbüchern zum Physikalischen Praktikum. Diese Bücher sind im internen Netz der UdS kostenlos zugänglich.

### 4.2 Fragen und Berechnungen

Beantworten Sie vor dem Versuchstermin folgende Fragen schriftlich in Ihrem Laborbuch:

- 1) Erklären Sie den Unterschied zwischen Analog- und Digitalelektronik.
- 2) Wie sind das binäre und das hexadezimale Zahlensystem aufgebaut? Drücken Sie Ihr Alter dezimal, binär und hexadezimal aus?
- 3) Wie viele Bit werden benötigt, um die Anzahl Tage eines Jahres benennen zu können?
- 4) Erklären Sie die Funktionsweise eines Bipolartransistors.
- 5) Erstellen Sie Wertetabellen für folgende logische Verknüpfungen mit jeweils 3 Eingängen: AND, NAND, OR, NOR, XOR.
- 6) Informieren Sie sich über die Funktionsweise und mögliche Schaltungen eines Halbaddierers und eines Volladdierers.

- [1] Th. Elze und C. Freudenberger, *Digital-Elektronik - Vorlesung für Studenten der Physik und Informatik*, J. W. Goethe-Universität Frankfurt am Main, 1998. (link zur Internetseite).
- [2] Hans Joachim Eichler, Heinz-Detlef Kronfeldt, Jürgen Sahm, *Das neue Physikalische Grundpraktikum*, Springer Spektrum, Berlin, Heidelberg, 2016.
- [3] Wolfgang Schenk, Friedrich Kremer, *Physikalisches Grundpraktikum*, Springer Spektrum Wiesbaden, 2014.

## 5 Zusatzmaterial

### 5.1 Schaltungsalgebra (Boole'sche Algebra)

Digitale Schaltungen sind aus wenigen logischen Grundschaltungen aufgebaut. Verknüpfungen und ganze Netzwerke aus diesen Grundschaltungen und die sich ergebenden Zusammenhänge beschreibt man mit Hilfe der Booleschen Algebra, die in der Digitaltechnik auch Schaltalgebra genannt wird. Ein digitales System arbeitet auf „binäre“ Art und Weise, d.h. es besteht aus Elementen, die nur zwei diskrete Zustände annehmen können, die als Null und Eins bezeichnet werden (Symbole „0“ und „1“). Im Falle eines mit elektronischen Schaltelementen realisierten digitalen Systems werden diese Werte den Schaltzuständen von Halbleiterschaltungen zugeordnet. Digitale Systeme lassen sich aber auch mit anderen eindeutigen diskreten Zuständen realisieren, z.B. „hell“ und „dunkel“, „Spin up“ und „Spin down“ usw. Das binäre Zahlensystem, das digitale Zustände beschreibt, und andere alternative Zahlensysteme können aufeinander abgebildet werden, siehe Tabelle.

Dezimal	Dual (binär)	Oktal	Hexadezimal
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Die Dezimalzahl  $19 = 1 \cdot 10^1 + 9 \cdot 10^0$  und die Binärzahl  $10011 = 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$  stellen beide denselben Zahlenwert dar.

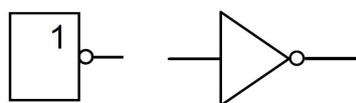
Eine binäre Ziffer nennt man *Bit*; eine Gruppe von 8 Bits wird als *Byte* bezeichnet und zwei Bytes bilden ein *Word*.

Es gibt drei logische Grundverknüpfungen, aus denen beliebig komplizierte Netze aufgebaut werden können: Konjunktion (UND), Disjunktion (ODER) und Negation (NICHT). In der Realisierung als elektronische Schaltung werden diese Verknüpfungen auch als „Gatter“ (englisch „Gate“) bezeichnet. Im Folgenden werden diese Verknüpfungen mit ihren Schaltsymbolen und ihren Wahrheitstabellen vorgestellt. Dabei bezeichnen A, B, ... jeweils die Eingänge und X, Y, Z die Ausgänge der Verknüpfung. Bei den Schaltsymbolen werden jeweils die Symbole nach DIN EN 60617-12 und die in den USA immer noch benutzten Symbole dargestellt. In der Schaltalgebra werden oft statt des Verknüpfungsopeators „^“ ein „•“ (UND-Verknüpfung, Boolesches Produkt) bzw. statt „V“ ein „+“ (ODER-Verknüpfung, Boolesche Summe) benutzt. Der „•“ wird oft weggelassen. Der Vorteil dieser Schreibweise ist, dass ohne großes Nachdenken die aus der normalen Algebra bekannten Rechenregeln benutzt werden können, z.B. Multiplikation vor Addition. Dies gilt jedoch nur, wenn in der benutzten Logik  $1 = \text{wahr}$  und  $0 = \text{falsch}$  sind.

#### NICHT-Verknüpfung

Die NICHT-Verknüpfung (auch Negation oder NOT) kehrt den Wert einer logischen Variablen um:

$$X = \bar{A} \text{ oder } X = \neg A$$



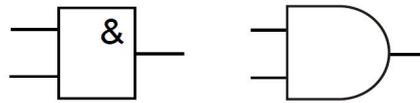
A	X
0	1
1	0

Symbol nach DIN EN 60617-12 (links) und USA-Symbol (rechts), sowie Wahrheitstabelle der NICHT-Verknüpfung.

**UND-Verknüpfung (Konjunktion)**

Bei einer UND-Verknüpfung (auch AND) nimmt die Ausgangsvariable dann und nur dann den Wert 1 an, wenn alle Eingangsvariablen den Wert 1 besitzen:

$$X = A \wedge B \text{ oder } X = A \bullet B$$



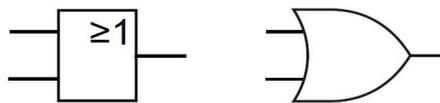
A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

Symbol nach DIN EN 60617-12 (links) und USA-Symbol (rechts), sowie Wahrheitstabelle der UND-Verknüpfung.

**ODER-Verknüpfung (Diskunktion)**

Bei einer ODER-Verknüpfung (auch OR) nimmt die Ausgangsvariable den Wert 1 an, wenn mindestens eine Eingangsvariable den Wert 1 besitzt:

$$X = A \vee B \text{ oder } X = A + B$$



A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

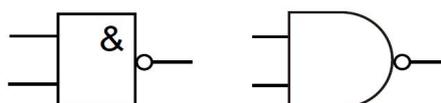
Symbol nach DIN EN 60617-12 (links) und USA-Symbol (rechts), sowie Wahrheitstabelle der ODER-Verknüpfung.

Mit den drei Verknüpfungen NICHT, UND und ODER können alle logischen Verknüpfungen dargestellt werden. Allerdings lassen sich mit realen Halbleiterschaltungen die abgeleiteten Verknüpfungen NAND und NOR einfacher realisieren. Mit diesen abgeleiteten Funktionen kann ebenfalls jede komplexe logischen Verknüpfungen konstruiert werden.

**NAND-Verknüpfung**

Der Name ist abgeleitet aus NOT und AND. Der Ausgang des Gatters ist 1, wenn mindestens einer der Eingänge nicht 1 ist:

$$X = \overline{A \wedge B} \text{ oder } X = \overline{A \bullet B}$$



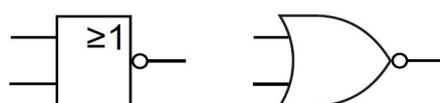
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

Symbol nach DIN EN 60617-12 (links) und USA-Symbol (rechts), sowie Wahrheitstabelle der NAND-Verknüpfung.

**NOR-Verknüpfung**

Der Name ist abgeleitet aus NOT und OR. Der Ausgang ist 1, wenn keiner der Eingänge 1 ist.

$$X = \overline{A \vee B} \text{ oder } X = \overline{A + B}$$



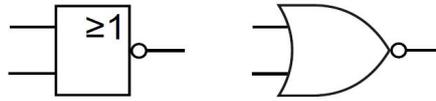
A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

Symbol nach DIN EN 60617-12 (links) und USA-Symbol (rechts), sowie Wahrheitstabelle der NOR-Verknüpfung.

### XOR-Verknüpfung (Antivalenz)

Der Name ist eine Abkürzung für „eXclusive OR“. Der Ausgang ist 1, wenn entweder am Eingang A oder am Eingang B der Wert 1 ist.

$$X = \overline{(A \wedge B) \vee (\bar{A} \wedge \bar{B})} \text{ oder } X = \overline{(A \bullet B) + (\bar{A} \bullet \bar{B})}$$



A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

Symbol nach DIN EN 60617-12 (links) und USA-Symbol (rechts), sowie Wahrheitstabelle der XOR-Verknüpfung.

Die Vereinfachungsregeln aus der Zahlenalgebra, dass Punktrechnung vor Strichrechnung geht, gilt analog in der Schaltalgebra, hier geht UND vor ODER. Für die Kammersetzung gelten folgende Regeln: Für die NOR-Verknüpfung und UND-Verknüpfung sind keine Klammern erforderlich. Für ODER-Verknüpfungen sind Klammern unbedingt erforderlich, damit die Bedeutung der Gleichung erhalten bleibt. Darüber hinaus gilt ein Satz von Axiomen (Peano Axiome: Kommutativgesetz, Assoziativgesetz, De Morgansche Gesetze, usw.), die Sie beispielsweise unter dem Stichwort „Boolesche Algebra“ in Wikipedia nachlesen können.

## 5.2 Physikalische Realisierung von Schaltfunktionen

Eine mögliche Darstellungsart der beiden logischen Zustände 0 und 1 ist die Zuordnung zu zwei verschiedenen Spannungswerten. Falls die 1 dem positiven Spannungspegel entspricht, gebraucht man den Begriff „positive Logik“; „negative Logik“ ordnet der 0 den positiveren Spannungspegel zu. Die absoluten Werte sind in keiner Weise maßgebend, insbesondere muss die 0 nicht einem Pegel von null Volt entsprechen. Bei der physikalischen Realisierung von Gattern benutzt man üblicherweise Spannungs- und stromgesteuerte Bauelemente, wie Dioden und Transistoren. Die Parameter einer solchen Anordnung, wie z.B. die Dioden-Schwelspannung oder der Gleichstromverstärkungsfaktor eines Transistors, sind von Bauteil zu Bauteil verschieden und ändern sich außerdem mit der Temperatur. Deshalb sind die Spannungspegel logischer Zustände nur innerhalb gewisser Toleranzgrenzen definiert, z.B.  $0 = 5 \text{ V} \pm 0.5 \text{ V}$ ;  $1 = -12 \text{ V} \pm 1 \text{ V}$ . Durch den Einsatz eines Transistors als aktives Bauelement werden am Ausgang Pegel von praktisch 0 Volt und 5 Volt erreicht. Mit einem Transistor lässt sich als einfachstes logisches Bauelement ein Inverter realisieren.

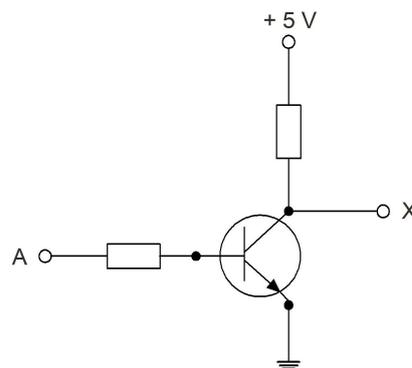


Abbildung 1: Inverter, bestehend aus einem Bipolartransistor und entsprechenden Vorwiderständen.

Heutige hochintegrierte Schaltkreise können auf einer Fläche von wenigen  $\text{cm}^2$  aus vielen Millionen Transistoren bestehen. Der Intel 6-Kern-Prozessor „Core i7“ (Sandy Bridge) hat 2.27 Milliarden Transistoren auf einer Fläche von  $431 \text{ mm}^2$ . Dies ist nur möglich, wenn die Größe der einzelnen Transistoren und ihre Leistungsaufnahme extrem verringert wird, was durch den Einsatz von MOSFETs und insbesondere der komplementären MOS-Technik (CMOS) gelingt.