

Computerlogik – Reader

2.) Umwandlung von Zahlen: Dezimalsystem ↔ binäres System

Das Dezimalsystem besteht aus zehn Ziffern: 1,2,3,4,5,6,7,8,9 und 0. Mit diesen Ziffern lassen sich jegliche Zahlen schreiben, indem man sie in der gewünschten Reihenfolge anordnet.

Das Binärsystem hingegen besteht nur aus zwei Ziffern: 1 und 0. Anders ausgedrückt: Ein oder Aus.

Gerechnet wird beim Binärsystem nicht wie beim Dezimalsystem in einer Schritten, sondern immer verdoppelnd. Begonnen wird mit 1 anschließend 2, dann 4, 8, 16, 32, 64, 128, ... , 65535. Geschrieben wird dabei von rechts nach links.

256	128	64	32	16	8	4	2	1
-----	-----	----	----	----	---	---	---	---

Umrechnung in das Binärsystem:

Eine Zahl X soll in das Binärsystem umgerechnet werden.

Zunächst sucht man sich die nächst kleinere Zahl im Binärsystem und bildet die Differenz. Gleichzeitig notiert man sich eine 1. Sollte die Differenz kleiner sein als die nächste Stelle im Binärsystem wird eine 0 notiert. Dieses Verfahren wird so lange wiederholt, bis die Zahl $X=0$ ist. Das folgende Beispiel soll dies noch einmal verdeutlichen.

Die Zahl 365 soll in das Binärsystem umgerechnet werden:

Die nächst kleinere Zahl von 365 ist im Binärsystem **256**. Also ist die letzte Ziffer eine 1.

Die Differenz von 365 zu 256 beträgt 109.

Die **128** wird also ausgelassen, weshalb die nächste Ziffer eine 0 ist.

$109-64=45$. Man notiert sich eine 1.

$45-32=13$, wieder notiert man sich eine 1. Die **16** wird ausgelassen, also 0.

$13-8=5$, erneut eine 1.

$5-4=1$, eine 1. Die **2** wird ausgelassen, also 0.

Zuletzt $1-1=0$, also eine 1.

256	128	64	32	16	8	4	2	1
1	0	1	1	0	1	1	0	1

$365=101101101$

Umrechnung in das Dezimalsystem:

Die Zahl 101101101 soll in das Dezimalsystem umgerechnet werden.

Die jeweilige Ziffer im Binärsystem wird mit der Zahl aus dem Dezimalsystem multipliziert:

$1*256, 0*128, 1*64, 1*32, 0*16, 1*8, 1*4, 0*2, 1*1$.

Die Ergebnisse werden anschließend miteinander addiert: $256+64+32+8+4+1=365$

Noch einmal zur Verdeutlichung:

256	*	1=	256
128	*	0=	0
64	*	1=	64
32	*	1=	32
16	*	0=	0
8	*	1=	8
4	*	1=	4
2	*	0=	0
1	*	1=	1
			365

3.) Wahrheitstabellen für Logikgatter

Wahrheitstabelle – AND

Bei einer AND-Schaltung ist das Ergebnis $A \wedge B$ (gesprochen „A und B“) genau dann wahr, sprich 1, wenn A und B ebenfalls wahr sind. Ist mindestens eines davon falsch, also 0, dann ist $A \wedge B$ ebenfalls falsch.

A	B	$A \wedge B$
1	1	1
1	0	0
0	1	0
0	0	0

Wahrheitstabelle – OR

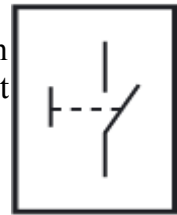
Bei einer OR-Schaltung ist das Ergebnis $A \vee B$ (gesprochen „A oder B“) wahr, wenn mindestens eine der Aussagen wahr ist. Erst wenn beide Aussagen falsch sind ist auch das Ergebnis falsch.

A	B	$A \vee B$
1	1	1
1	0	1
0	1	1
0	0	0

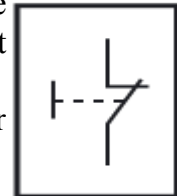
4.) Transistor als Schalter

Ein Transistor kann als Schalter fungieren. Dabei gibt es zwei Zustände: Sperrender Transistor-geöffneter Schalter und leitender Transistor-geschlossener Schalter. Der Schalter bei einem Transistor ist die Kollektor-Emitter-Strecke und die Basisstrecke steuert diese.

Zunächst wird der sperrende Transistor in seiner Funktion beschrieben.
 Fließt kein Strom durch die Basis, so kann auch kein Strom vom Kollektor zum Emitter fließen. Das bedeutet, dass der Steuerungsstrom dem Schalter nicht erlaubt sich zu öffnen. Der Transistor wirkt hierbei als ein hoher Widerstand.

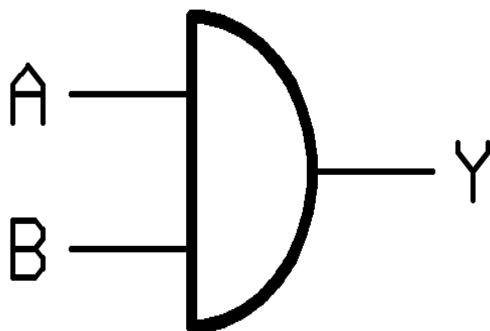


Nun zur Funktionsweise eines leitenden Transistors.
 Durch die Basis fließt ein Strom und erlaubt somit der Emitter-Kollektor-Strecke ebenfalls einen Stromfluss. Das bedeutet, dass der Transistor leitet und somit der Schalter geschlossen ist.
 Der Transistor ist auch in diesem Fall noch ein Widerstand, jedoch ein sehr geringer.



5.) Schaltplan: AND – Gatter mit Transistor

And-Gatter auch Und-Schaltungen genannt sind aus der heutigen Welt der Technik nicht mehr weg zu denken. Und-Schaltungen bestehen meistens aus zwei Eingängen (in diesem Fall A und B) und einem Ausgang (Y).



A	B	A und B
1	1	1
1	0	0
0	1	0
0	0	0

Das Schaltsymbol für ein Und-Gatter sieht so aus wie in der Abbildung zu sehen. Rechts daneben befindet sich eine Wahrheitstabelle. Sie gibt an, ob an dem Ausgang (Y) eine Spannung messbar ist. Das Besondere an einer Und-Schaltung ist, dass sowohl A als auch B aktiviert sein müssen um eine Spannungsmessung in Y zu ermöglichen. Der Wahrheitstabelle gibt genau das wieder.

Anwendung von And-Gattern mit Transistoren

Kombiniert man eine solche eben dargestellte Und-Schaltung mit Transistoren, bringt man die Möglichkeiten auf ein ganz neues Level. Dieses System wird häufig für Sensoren aller Art angewandt. In dem unten dargestellten Schaltkreis sieht man, wie zwei Transistoren in einer Und-Schaltung verbaut sind. Das heißt, die LED- fängt erst an zu leuchten, wenn beide Schalter aktiviert und somit der Kontakt von Kollektor zu Emitter geschlossen ist erst wenn dies bei beiden Transistoren der Fall ist fängt die LED an zu leuchten. Es handelt sich also um eine Und-Schaltung.

Besonders interessant wird es allerdings erst, wenn wir uns vorstellen, dass die beiden Schalter (S1 und S2) durch Sensoren ersetzt werden. S1 sei eine Zeitschaltuhr, die von 20-6 Uhr aktiviert ist und S2 sei ein Bewegungssensor. Fast vor jedem Haus ist eine solche Und-Schaltung installiert. Die LED würde also erst anfangen zu leuchten, wenn es zwischen 20 und 6 Uhr ist **und** wenn eine Person den Bewegungssensor auslöst. Das funktioniert dann folgendermaßen: Der Schaltkreis besitzt zwei Spannungsquellen. Die Eine versorgt die beiden Sensoren und die andere befindet sich in einem weiteren Stromkreis mit der LED. Das bringt einen weiteren Vorteil mit sich, nämlich den, dass die Sensoren häufig eine andere Spannung als die LED benötigen. Wenn nun die Schalter (Sensoren) aktiviert werden fließt der Strom weiter zur Basis des Transistors und dieser schließt die Verbindung von Emitter zu Kollektor sodass sich der Stromkreis schließt und die LED anfängt zu leuchten.

