

B1

"Wahr" ~ "1" ~ "high" ~ "H" ~ "spannig"

Tatsächlich basiert die ganze Logik und Arithmetik in einem Computer auf unseren wenigen logischen Grundfunktionen. Diese müssen aber hardware-mäßig realisiert werden.

Diese Logik-Schaltungen liegen auf der Grenze zwischen Mathematik und Physik oder zwischen der Logik und der Elektronik.

Die verwendeten Schaltzeichen stellen genauso eine elektronische Komponente wie eine logische Funktion dar als auf ein Bauteil, das man kaufen kann!

Mit diesen "Logikgattern" kann sich gleichzeitig komplexe logische Abhängigkeiten darstellen und das Verdrahtungsschema von elektronischen Komponenten aufzeichnen.

Die Inputs (Wahrheitswerte, Daten) werden elektrisch durch Spannungen realisiert (~~und nicht durch~~ ^{statt z.B.} mechanisch durch irgendwelche Hebelstellungen ...). Sie müssen ^{sich} mit der folgenden äquivalenten Darstellung befassen:

Logik-Schreibweise		wahr		falsch
Digitale Schreibweise		1		0
Elektronik-Schreibweise	high	H		L low
Spannungsspekt *		< 0,8 V (max < 0,1 V)		> 2,8 V (max > 4,8 V)

*) gilt für 5 Volt - TTL-Bausteine, eine schon etwas belastete, aber immer noch erhältliche Bausteinserie.

Die Input-Leitungen sind immer links, der Output ist rechts angeordnet. Jeweill sollen in den Schaltungen die Informationen / Signale vom irgend möglich von links nach rechts fließen. Die folgende Tabelle gibt die Logikgatter mit den modernen Ekt-Symbolen. (In der USA werden immer noch andere, weniger eingängige Symbole verwendet.) Sie mache 2 einfache Beispiele.

~~B3~~

Tabelle der Logik-Gatter


Die Kopie stellt alle Logik-Glieder zusammen, die wir verwenden.

Links : Der Name der Logik Funktion

Mitte : Das Schaltzeichen (moderne EU-Variante)

Rechts : Funktionschritte, Wiederholung der Definition

Die Informations flüsse immer von links nach rechts durch diese Schaltglieder. Die Anschlüsse links sind also Input-Leitungen, der Anschluss rechts ist die Output-Leitung.

Ein  vor dem Output bedeutet die Negation des ganzen Symbols (vgl. OR/NOR, AND/NAND, XOR/EXOR, 1/NOT)

Im nächsten Abschnitt (B4) lernen wir, Schaltungen aus mehreren solchen Gliedern zu analysieren.

