

# Halbaddierer

Für die ersten Überlegungen vereinfachen wir unser Additionsproblem auf einstellige Binärzahlen:



Wir geben zwei 1-Bit Zahlen zur Addition ein und erhalten die Summe - oder einen Übertrag (Carry), wenn die Summe 2 ist.

Wir können als Tabelle aufschreiben, was unsere Schaltung tun soll:

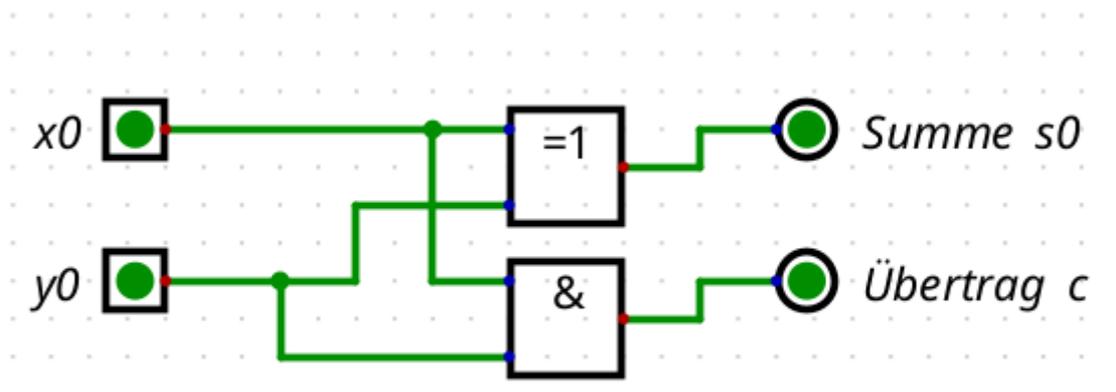
$x_0$	$y_0$	$s_0$	$c$
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Hieraus kann man zwei logische Funktionen ablesen, eine für den Übertrag und eine für die Summe:

- Summe:  $s_0 = (x_0 \vee y_0) \wedge \neg(x_0 \wedge y_0)$
- Übertrag:  $c = x_0 \wedge y_0$

Die Summe  $s_0$  ist also die XOR Verknüpfung von  $x_0$  und  $y_0$ , der Übertrag  $c$  die UND Verknüpfung von  $x_0$  und  $y_0$

Mit diesen Erkenntnissen können wir nun einen **Halbaddierer** konstruieren. Ein Halbaddierer kann zwei 1-Bit Zahlen korrekt addieren, berücksichtigt jedoch nicht, ob bei der Addition ein Übertrag aus einem vorigen Schritt zu beachten ist.





## (A1)

Baue in deiner Simulationssoftware einen Halbaddierer und speichere diesen als neues Bauteil ab.

---

Um den Übertrag berücksichtigen zu können, wenden wir und im nächsten Schritt dem [Volladdierer](#) zu.

From:  
<https://info-bw.de/> -

Permanent link:  
<https://info-bw.de/faecher:informatik:oberstufe:techninf:logikschaltungen:digitaltechnik:addierer:halbaddierer:start>

Last update: **19.10.2022 20:29**

