

Schaltungen

- Untersuche die Addition von zwei einstelligen Binärzahlen a und b
 - Erstelle eine Wahrheitstabelle mit je einer Spalte für die Ergebnisstelle und den auftretenden Übertrag
 - Untersuche, welche aussagenlogische Verknüpfung jeder Spalte zugrunde liegt

- Eine Schaltung, die zwei einstellige Binärzahlen addiert, nennt man

Halbaddierer

- Eingänge: Einstellige Binärzahlen a, b
- Ausgänge: Ergebnisstelle s , Übertrag c_{out} (carry out)

- Realisiere mit Logisim einen Halbaddierer
 - Verwende nur die Gatter UND, ODER, NICHT

- Wie addieren wir mehrstellige Binärzahlen?
- Idee: Ein Halbaddierer pro Stelle
 - Problem: Was ist mit dem Übertrag aus der Addition der vorherigen Stelle?

- Untersuche die Addition von drei einstelligen Binärzahlen a , b und c_{in}
 - Erstelle eine Wahrheitstabelle mit je einer Spalte für die Ergebnisstelle und den auftretenden Übertrag
 - Untersuche, welche aussagenlogische Verknüpfung jeder Spalte zugrunde liegt

- Eine Schaltung, die drei einstellige Binärzahlen addiert, nennt man

Volladdierer

- Eingänge: Einstellige Binärzahlen a , b , c_{in}
- Ausgänge: Ergebnisstelle s , Übertrag c_{out}

Wahrheitstabelle Volladdierer

c_{in}	a	b	s	c_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Es gilt:

- $s = (a \text{ XOR } b) \text{ XOR } c_{\text{in}}$
- $c_{\text{out}} = (a \wedge b) \vee (c_{\text{in}} \wedge (a \text{ XOR } b))$

Herleitung c_{out} (1)

- Betrachte nur die Zeilen der Wahrheitstabelle, für die c_{out} den Wert 1 annimmt:

c_{in}	a	b	c_{out}
0	1	1	1
1	0	1	1
1	1	0	1
1	1	1	1

- Bilde daraus die sogenannte **Disjunktive Normalform** (DNF):

$$c_{\text{out}} = (\neg c_{\text{in}} \wedge a \wedge b) \vee (c_{\text{in}} \wedge \neg a \wedge b) \vee (c_{\text{in}} \wedge a \wedge \neg b) \vee (c_{\text{in}} \wedge a \wedge b)$$

- Alle mit 0 belegten Eingangsvariablen werden negiert, alle mit 1 belegten Eingangsvariablen nicht

Herleitung c_{out} (2)

- Zur besseren Lesbarkeit:
 - Weglassen von \wedge (ähnlich wie \cdot bei einer Multiplikation)
 - Weglassen unnötiger Klammern (\wedge vor \vee)
 - Negation durch Überstrich statt \neg

$$c_{\text{out}} = \overline{c_{\text{in}}}ab \vee c_{\text{in}}\overline{a}b \vee c_{\text{in}}a\overline{b} \vee c_{\text{in}}ab$$

- Umsortieren der Terme:

$$c_{\text{out}} = \overline{c_{\text{in}}}ab \vee c_{\text{in}}ab \vee c_{\text{in}}a\overline{b} \vee c_{\text{in}}\overline{a}b$$

- Ausklammern:

$$c_{\text{out}} = ab(\overline{c_{\text{in}}} \vee c_{\text{in}}) \vee c_{\text{in}}(a\overline{b} \vee \overline{a}b)$$

- Aktueller Ausdruck:

$$c_{\text{out}} = ab(\underbrace{\overline{c_{\text{in}}} \vee c_{\text{in}}}_{=1}) \vee c_{\text{in}}(\underbrace{a\overline{b} \vee \overline{a}b}_{=a \text{ XOR } b})$$

- Vereinfachen:

$$c_{\text{out}} = ab \vee c_{\text{in}}(a \text{ XOR } b)$$

- Wieder ausführlich geschrieben:

$$c_{\text{out}} = (a \wedge b) \vee (c_{\text{in}} \wedge (a \text{ XOR } b))$$



- Realisiere mit Logisim einen Volladdierer
 - Die Verwendung eines XOR-Gatters ist erlaubt

- Realisiere mit Logisim einen Addierer für Binärzahlen mit mindestens 4 Stellen
 - Verwende Deinen eigenen Volladdierer (aus Aufgabe 4)