1.4.5. Video zeigen, wie logisches Und (= Konjunktion), Oder (= Disjunktion) und Nicht (= Negation) elektronisch realisiert werden können: Sebastian Lague: Exploring How Computers Work, https://www.youtube.com/watch?v=QZwneRb-zqA bis 6:16.

Disjunktive Normalform (DNF)

Definition 1.4.6

Eine **Wahrheitswertefunktion** oder **Boolesche Funktion** ist eine Funktion, deren Inputs Wahrheitswerte sind und die als Output einen Wahrheitswert liefert.

1.4.7. Jede Wahrheitstafel stellt eine boolesche Funktion dar. Jeder logische (= boolesche) Term (etwa $a \wedge (b \vee \overline{a})$ stellt eine boolesche Funktion dar.

Satz 1.4.8 Disjunktive Normalform für boolesche Funktionen

Jede boolesche Funktion/Wahrheitstafel kann durch einen logischen (= booleschen) Term beschrieben werden, ja sogar durch einen logischen Term in **disjunktiver Normalform** (**DNF**), d. h. durch eine «Ver-Oder-ung (= Disjunktion) von Ver-Und-ungen der Inputs bzw. deren Verneinungen».

Insbesondere ist jeder boolesche Term zu einem Term in disjunktiver Normalform äquivalent.

Beweis. Wir erklären das allgemeine Verfahren «Bilden der **disjunktiven Normalform**» an einem aussagekräftigen Beispiel (mit drei Inputs).

a	b	c	
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Der folgende logische Ausdruck löst also unser Problem:

□

⚒ **Aufgabe A20** Betrachte die Wahrheitswertefunktion $f = f(a, b, c)$ mit drei Inputs a, b und c, die genau dann 1 liefert, wenn genau zwei der drei Inputs 1 sind.

Finde einen booleschen Ausdruck in disjunktiver Normalform für diese Funktion.

Schaltungen bauen mit Logik-Simulations-Software (etwa Logisim)

1.4.9. Installiere Logisim über den folgenden Link (vermutlich wirst du ausserdem Java installieren müssen - während der Logisim-Installation wirst du hoffentlich auf die entsprechende Java-Installations-Webseite geleitet): https://sourceforge.net/projects/circuit/

(eventuell meine alten Videos erwähnen)

Aufgabe A21 Baue mit Logisim das Bauteil «XOR» (= exklusiver Oder = Entweder-Oder). Es hat zwei Eingänge/Inputs x und y und einen Ausgang/Output. Der Output wird genau dann 1, wenn genau einer der Inputs 1 ist.

Empfohlenes Vorgehen:

- Erstelle zunächst die gewünschte Wahrheitstabelle (zwei Spalten für die Inputs, eine Output-Spalte).
- Finde einen geeigneten logischen Ausdruck für den Output (etwa per disjunktiver Normalform).
- Realisiere das Bauteil in Logisim und nenne es XOR.

Aufgabe A22 Halbaddierer: Addition zweier Bits = einstelliger Binärzahlen

Baue mit Logisim das Bauteil «HA» = «Halbaddierer». Es hat zwei Eingänge/Inputs x und y und zwei Ausgänge s und c.

Wenn man die beiden Inputs x und y als einstellige Binärzahlen interpretiert, so sollen die beiden Output-Bits c und s nebeneinandergeschrieben die Summe $x + y$ darstellen. Beispielsweise soll bei den Inputs $x = 1$ und $y = 1$ als Output $c = 1$ und $s = 0$ herauskommen, denn binär gilt $1 + 1 = 10$.

Bemerkung: s steht für *sum*=Summe, c für carry (bit)=Übertrag.

Empfohlenes Vorgehen:

- Erstelle zunächst die gewünschte Wahrheitstabelle (zwei Spalten für die Inputs, zwei Spalten für die Outputs).
- Finde geeignete logische Ausdrücke für die beiden Outputs.
- Realisiere das Bauteil in Logisim.

Aufgabe A23 Volladdierer: Addition dreier Bits = einstelliger Binärzahlen

Baue mit Logisim das Bauteil «VA» = «Volladdierer». Es hat drei Eingänge/Inputs x, y, z und zwei Ausgänge s und c.

Wenn man die drei Inputs x, y, z als einstellige Binärzahlen interpretiert, so sollen die beiden Output-Bits c und s nebeneinandergeschrieben die Summe $x + y + z$ darstellen. Beispielsweise soll bei den Inputs $x = 1$ und $y = 1$ und $z = 1$ als Output $c = 1$ und $s = 1$ herauskommen, denn binär gilt $1 + 1 + 1 = 11$.

Empfohlenes Vorgehen:

- Erstelle zunächst die gewünschte Wahrheitstabelle (drei Input-Spalten, zwei Output-Spalten).
- Realisiere das Bauteil in Logisim. Es gibt mindestens drei sinnvolle Lösungswege dafür:
  - Verwende zwei Halbaddierer und ein weiteres Bauteil.
  - Finde einen logischen Ausdruck für jeden Output durch folgende Überlegungen:
    - s wird genau dann 1, wenn «ungeradzahlig viele» der Inputs 1 sind (dies klingt nach XOR).
    - c wird genau dann 1, wenn mindestens zwei der Inputs 1 sind.
  - Finde einen logischen Ausdruck für jeden Output per disjunktiver Normalform.

1.4.10. Wenn man an einen Voll-Addierer an einen der drei Eingänge konstant den Wert 0 legt, erhält man einen Halbaddierer.

Aufgabe A24 Baue nun einen 4-Bit-Addierer mit Logisim. Ein 4-Bit-Addierer addiert zwei 4-stellige Binärzahlen und hat

- 8 Inputs x_3, x_2, x_1, x_0 und y_3, y_2, y_1, y_0
- 5 Outputs s_4, s_3, s_2, s_1, s_0

Der Zusammenhang zwischen Inputs und Outputs ist durch die folgende «binäre Gleichung» gegeben

$$x_3x_2x_1x_0 + y_3y_2y_1y_0 = s_4s_3s_2s_1s_0 \qquad \text{bzw. gleichbedeutend} \qquad \begin{array}{rr} & x_3x_2x_1x_0 \\ + & y_3y_2y_1y_0 \\ \hline & s_4s_3s_2s_1s_0 \end{array}$$

wobei der «x-Input» ebenso wie der «y-Input» als 4-stellige Binärzahl und der «s-Output» als 5-stellige Binärzahl aufzufassen sind. Schliesse «x-Input», «y-Input» und «s-Output» jeweils an eine Hexadezimalanzeige an, damit du die Rechnungen rasch überprüfen kannst.

Hinweis: In der schriftlichen Addition sind ein Halbaddierer und drei Volladdierer «versteckt».