

Binär codierte Daten vergleichen und addieren

Äquivalenzschaltung

Es gibt Programme, mit denen man logische Schaltungen simulieren kann. Die logischen Grundfunktionen können mit Hilfe einer geeigneten Software ausprobiert werden. Ein Beispiel ist die Software LogicSim. LogicSim ist Freeware. Es handelt sich um eine Java-Applikation; Java muss also auf dem PC installiert sein, um LogicSim nutzen zu können (18\18-materialien\tools\LogicSim\LogicSim.jar). Die neueste Version unter <http://www.tetzi.de/>.

1. Nach dem Programmstart steht ein neuer Funktionsplan zur Verfügung. Ein Objekt erstellt man, indem man die erforderliche Klasse mit dem Mauszeiger auswählt und auf dem Funktionsplan ablegt.

2. Z. B. für eine UND-Schaltung werden zwei Objekte der Klasse Schalter (E1 und E2) benötigt, ein AND-Gatter (&) und eine LED-Anzeige (A)

3. Zur Verbindung von Objekten muss der Ausgang eines Objekts angeklickt werden. Durch Anklicken des Eingangs des zu verbindenden Objekts wird eine Leitung gezogen.

4. Nach Start der Simulation können die Schalter E1 und E2 durch Anklicken betätigt und das Ergebnis an der LED A beobachtet werden.

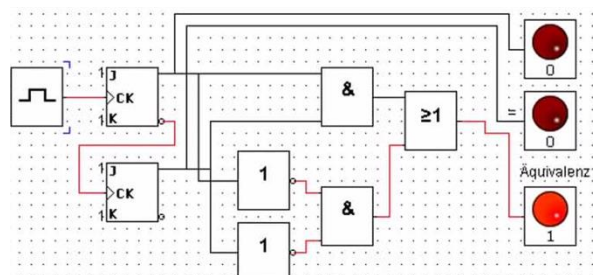
Zu den Schaltungen sind hier auch Animationen verfügbar. Du kannst nachher entweder die Animationen verwenden oder die Schaltungen in dem Programm LogicSim öffnen und die Simulation starten.

Die erste Schaltung ist eine *Äquivalenzschaltung*. Damit kann festgestellt werden, ob zwei Bit gleich sind („vergleichen“). Da die Schaltungen hier mit Hilfe eines Taktgebbers (*Clock*) alle möglichen Eingangssignale durchlaufen, wird der aktuelle Zustand der Eingänge mit Hilfe zweier LEDs angezeigt.

- Öffne die Animation der Äquivalenzschaltung (des „Vergleichers“) in der Datei 18\18-materialien\animationen\vergleich\vergleich-binaerzaehler.htm oder im Programm LogicSim die Schaltung *vergleich.lsim*, um die Äquivalenzschaltung zu simulieren.

Fülle die Wertetabelle dazu aus.

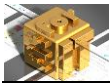
E1	E2	Äquivalenz
0	0	1
0	1	0
1	0	0
1	1	1



- Beschreibe das Ergebnis der Simulation.

Der Ausgang wird immer auf 1 gesetzt („wahr“), wenn die beiden Eingänge gleich sind.

- Die Schaltung $E1 \wedge E2 \vee \overline{E1} \wedge \overline{E2}$ („E1 UND E2 ODER NICHT E1 UND NICHT E2“) wird deshalb auch als **Äquivalenzschaltung** bezeichnet.



Halbaddierer

George Boole entwickelte Mitte des 19. Jahrhunderts die Grundlagen der nach ihm benannten *Booleschen Algebra*. Diese Arbeit ermöglichte es, Rechenverfahren im Dualsystem maschinell mit Hilfe der binären Logik auszuführen.

Das einfachste Beispiel ist die **Addition zweier einstelliger Dualzahlen**:

Dabei gibt es insgesamt nur vier Möglichkeiten:

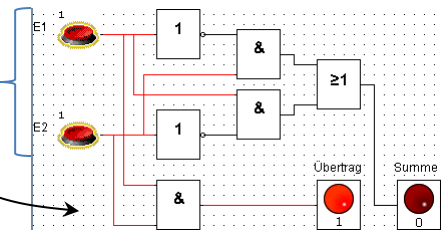
0	0	1	1	← E1
+0	+1	+0	+1	← E2
0	1	1	0	← A

Wenn man jetzt die beiden Summanden als Eingang einer logischen Schaltung sieht und den Übertrag sowie die Summe als Ausgang, ergibt sich die folgende Wertetabelle:

E1	E2	Übertrag	Summe
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Das Prinzip dieser Schaltung ist folgendes: Der Ausgang wird immer auf 1 gesetzt („wahr“), wenn die beiden Eingänge *ungleich* sind. Es liegt also das *Gegenteil einer Äquivalenzschaltung* vor, eine **Antivalenzschaltung**.

Die Schaltung für den Übertrag ist ein einfaches UND. Wenn dafür eine zweite LED in die Antivalenzschaltung eingesetzt wird, kann man mit der Schaltung zwei einstellige Dualzahlen addieren (siehe Abbildung rechts). Diese Schaltung wird **Halbaddierer** genannt.



- Öffne die Animation des Halbaddierers in der Datei 18\18-materialien\animationen\halbaddierer\halbaddierer-binaerzaehler.htm oder im Programm LogicSim die Schaltung halbaddierer.lsim, um den Halbaddierer zu simulieren.

Überprüfe, ob die Simulation die geforderten Ergebnisse erzielt, indem du die Wertetabelle ausfüllst und mit der Wertetabelle oben vergleichst.

E1	E2	Übertrag	Summe
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

- Gib die Funktionsterme der beiden Schaltungen an.

Übertrag: $E1 \wedge E2$

Summe: $\overline{E1} \wedge E2 \vee E1 \wedge \overline{E2}$