

Wie addiert der Computer?

Um herauszufinden, wie ein Schaltnetz für die Addition von Binärzahlen aussehen könnte, wiederholen wir die schriftliche Addition von Binärzahlen und beobachten dabei genau die einzelnen Schritte.

Aufgaben

1. Addiere zunächst die beiden Zahlen $101101b$ und $110101b$ schriftlich. Schreibe sie dazu sauber untereinander und notiere auch immer den Übertrag.
2. Kontrolliere deine Lösung, indem du beide Summanden und dein Ergebnis ins Dezimalsystem umwandelst und dort überprüfst.
3. Bei der Beispielrechnung war die Summe aus zwei 6-Bit-Zahlen eine 7-Bit-Zahl. Entscheide, ob auch 8- oder 9-Bit-Zahlen entstehen können und untersuche, aus wie vielen Bit die Summe zweier n -Bit-Zahlen maximal bestehen kann.
4. Die Addition der hintersten beiden Bits ist noch etwas einfacher als die Addition der weiteren Bits. Erläutere kurz, warum es sich ab dem zweiten Bit von hinten verändert.

Lösungen

1.

$$\begin{array}{r}
 101101b \\
 + 110101b \\
 \hline
 111101 \\
 \hline
 1100010b
 \end{array}$$

2.

$$\begin{array}{l}
 101101b = 1 + 4 + 8 + 32 = 45 \\
 110101b = 1 + 4 + 16 + 32 = 53 \\
 \left. \begin{array}{l} 45 \\ 53 \end{array} \right\} \text{Summe: } \underline{98} \\
 \\
 1100010b = 2 + 32 + 64 = \underline{98} \quad \text{😊}
 \end{array}$$

3. Bei der Addition zweier Binärzahlen gibt es an jeder Stelle drei Möglichkeiten: Wenn bei beiden Zahlen an der Stelle eine 0 steht, so ist das Bit in der Summe 0 und das Übertragsbit auch 0; bei einer 0 und einer 1, ist das Summenbit eine 1 und das Übertragsbit eine 0; steht bei beiden Zahlen eine 1, so erhält man als Summe $10b$ (2), also ist das Summenbit eine 0 und das Übertragsbit eine 1. All das gilt auch für das vorderste Bit. Deshalb kann die Summe zweier Binärzahlen maximal um ein Bit länger werden, wenn das letzte Übertragsbit eine 1 ist. Die Summe zweier n -Bit-Zahlen ist also maximal eine $(n+1)$ -Bit-Zahl.

RECHNER UND NETZE

4. An der letzten Stelle müssen nur zwei Bits addiert werden. Ab der zweiten Stelle von hinten kommt das Übertragsbit hinzu. Es müssen also drei Bits addiert werden.

Betrachten wir zunächst einmal zwei allgemeine vierstellige Binärzahlen:

$$\begin{array}{r}
 a_3 a_2 a_1 a_0 \\
 + \quad b_3 b_2 b_1 b_0 \\
 \hline
 \underline{\underline{ü_3 ü_2 ü_1 ü_0}} \\
 s_4 s_3 s_2 s_1 s_0
 \end{array}$$

Wir starten jetzt zuerst mit den beiden letzten Bits und überlegen uns alle Möglichkeiten, die vorkommen können.

5. Fülle die folgende Wahrheitstafel aus. Dabei sind a_0 und b_0 die letzten Bits aus der Rechnung von oben, s_0 ist das letzte Bit der Summe und $ü_0$ ist der Übertrag für die Addition der nächsten Bits (siehe oben).

| a_0 | b_0 | $ü_0$ | s_0 |
|-------|-------|-------|-------|
| 0 | 0 | | |
| 0 | 1 | | |
| 1 | 0 | | |
| 1 | 1 | | |

Wir suchen für die beiden Ausgänge s_0 und $ü_0$ mögliche Logikgatter, die wir nutzen können.

6. Gib jeweils das Logikgatter an, dem Ausgang s_0 und Ausgang $ü_0$ entsprechen.

Lösungen

5.)

| a_0 | b_0 | $ü_0$ | s_0 |
|-------|-------|-------|-------|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |

6.)

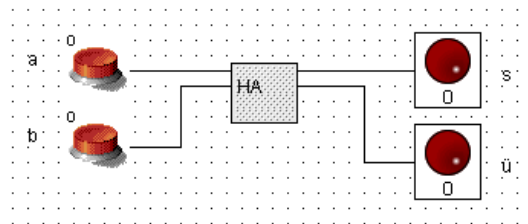
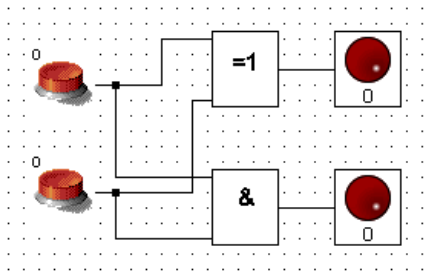
(AND-Gatter) $a_0 \wedge b_0$ $a_0 \oplus b_0$ (XOR-Gatter)

$0b + 0b = 0b$
 $0b + 1b = 1b$
 $1b + 0b = 1b$
 $1b + 1b = 10b$

7. Baue dir in LogicSim ein Schaltnetz, das für zwei Eingaben die beiden gewünschten Ausgaben besitzt und testet es. Erstelle daraus ein neues Modul und nenne es **Halbaddierer** (Kurzform für die Beschriftung: **HA**).

Lösung:

Die Anleitung zur Erstellung eines Moduls findest du in 02_run_anleitung_logic_sim.



...

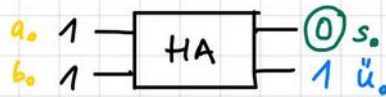
8. Probiere das ganze mit dem Rechenbeispiel aus Aufgabe 1 schrittweise aus. Überlege dir, welche Möglichkeiten für \ddot{u}^* und \ddot{u}° gegeben sind und welcher Wert letztendlich bei \ddot{u}_1 herauskommen muss.
9. Gib ein Gatter an, dem der erwünschte Ausgang \ddot{u}_1 mit den Eingängen \ddot{u}^* und \ddot{u}° entspricht.
10. Entwerfe und ergänze das Schaltnetz von oben und teste es für alle Schritte der Aufgabe 1.
11. Erstelle aus dem Schaltnetz aus Aufgabe 10 ein neues Modul mit drei Eingängen und zwei Ausgängen und nenne es **Volladdierer** (Kurzform für die Beschriftung: **VA**).
12. Überlege dir jetzt, wie du aus den neuen Modulen (Halbaddierer und Volladdierer) ein Schaltnetz für die gesamte Addition aus Aufgabe 1 entwerfen kannst. Übertrage deine Überlegungen in eine LogicSim-Konstruktion und teste sie mit Aufgabe 1 und weiteren Aufgaben aus.

Lösungen:

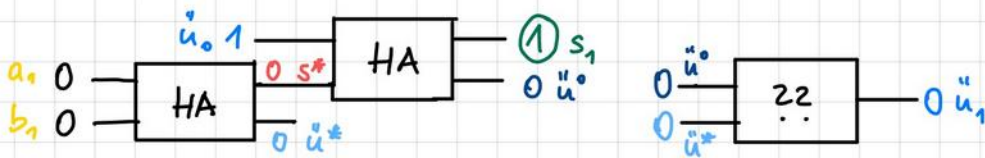
8.)

$a_5 a_4 a_3 a_2 a_1 a_0$
 $1 0 1 1 0 1$
 $b_5 b_4 b_3 b_2 b_1 b_0$
 $1 1 0 1 0 1$

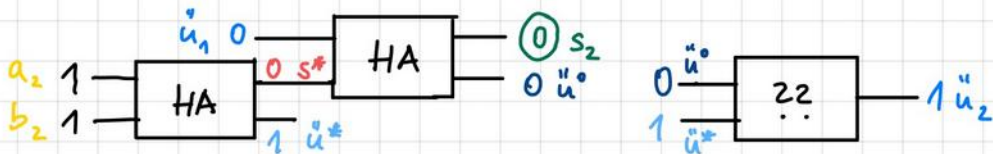
Schritt 1:



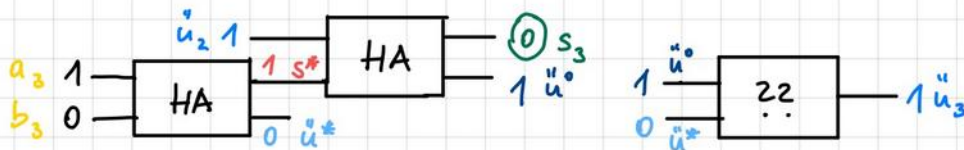
Schritt 2:



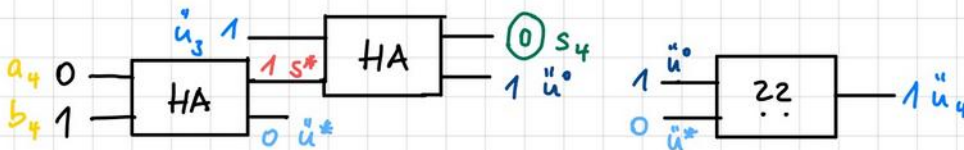
Schritt 3:



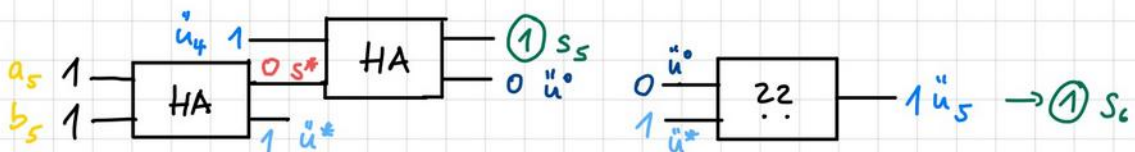
Schritt 4:



Schritt 5:

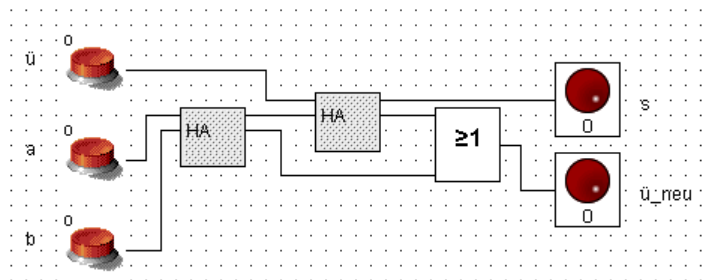
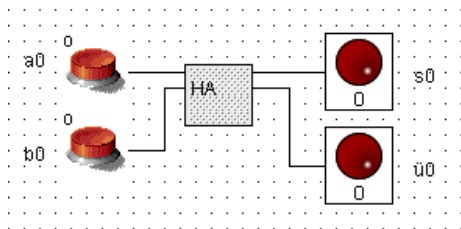


Schritt 6:



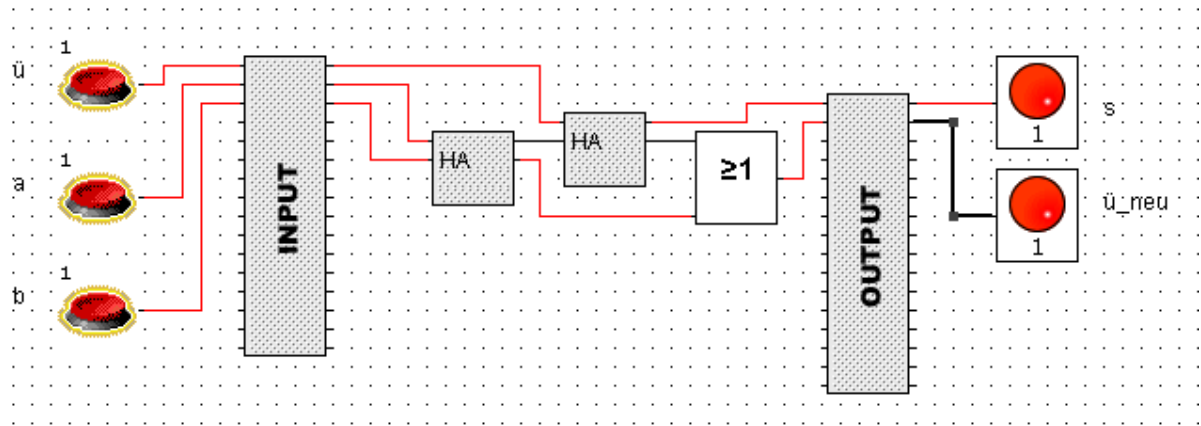
9. Der letzte mögliche Fall, dass \ddot{u}^* und \ddot{u}° beide 1 sind, kann nicht auftreten. Damit lässt sich die Disjunktion nutzen, um das endgültige Übertrags-Bit herauszubekommen.

10. Schritt 1 und alle weiteren Schritte:



Beim Simulieren in LogicSim sieht man, dass das Schaltnetz zur Rechnung passt.

11. Die Anleitung zur Erstellung eines neuen Moduls findet man im Dokument 02_run_anleitung_logicsim.



12. Achtung: Das Ergebnis muss von unten nach oben gelesen werden.

