

6. Übungsblatt - zu bearbeiten bis 4.12.2023**Aufgabe 1** Arithmetik: Addition und Subtraktion

- a) Wandeln Sie die Zahl 1101100111110110_2 ins Hexadezimalsystem und die Zahl $4AC9E_{16}$ ins Binärsystem um!
- b) Gegeben seien die beiden vorzeichenlos zu interpretierenden Binärzahlen

$$a = 00101_2 \quad \text{und} \quad b = 11101_2.$$

Berechnen Sie $a + b$ und $a - b$ im Binärsystem!

Überprüfen Sie Ihre Rechnung im Dezimalsystem!

- c) Gegeben seien die beiden vorzeichenlos zu interpretierenden Binärzahlen

$$a = 1010011_2 \quad \text{und} \quad b = 1011011_2.$$

Berechnen Sie $a + b$ und $a - b$ im Binärsystem!

Überprüfen Sie Ihre Rechnung im Dezimalsystem!

Aufgabe 2 Arithmetik: Halbsubtrahierer

Bei der Subtraktion zweier Binärziffern (Bits) entsteht u.U. ein negativer Übertrag, den wir hier “Borger” nennen wollen.

- a) Geben Sie die Wertetabelle für den Halbsubtrahierer, der zwei Binärziffern voneinander abzieht, mit den Ausgängen Differenz D und Borger B an!
- b) Geben Sie die Booleschen Funktionen für Differenz D und Borger B an!
- c) Zeichnen Sie unter Verwendung der Ihnen bekannten (Logik-)Gatter eine Schaltung, die den Halbsubtrahierer implementiert!

Aufgabe 3 Hardware-Beschreibungssprache (HDL)

Bearbeiten Sie diese Aufgabe mit dem Hardware-Simulator von der Webseite

<http://nand2tetris.org/software.php>

- a) Implementieren Sie einen Halb- und einen Volladdierer in HDL!
Zur Implementierung können Sie vorgebene logische Gatter wie z.B. NAND, NOR, AND, OR, NOT, XOR etc. verwenden (siehe `nand2tetris/tools/builtInChips`).
- b) Implementieren Sie einen 8-Bit-Übertragskette-Addierer in HDL!
(vgl. Kapitel 3, Folien zu „ n -Bit-Übertragskette-Addierer“)
- c) Implementieren Sie einen 8-Bit-Inkrementierer in HDL!
(Ein Inkrementierer erhöht seine Eingabe um 1.)
- d) Implementieren Sie einen 8-Bit-Übertragsauswahl-Addierer in HDL!
(vgl. Kapitel 3, Folien zu „Optimierung: n -Bit-Übertragsauswahl-Addierer“)