1 0	**Praxistutorium Grundzüge der Informatik B1**	**Aufgabenblatt 2**

Die folgenden Teilaufgaben bauen streng aufeinander auf. Sie sollten daher auch in der vorgegebenen Reihenfolge bearbeitet werden. Das Ziel ist die Implementation (und Simulation mit HADES) eines 4-Bit Addierers sowie eines 4-Bit Subtrahierers. Diese sind aus zuvor zu realisierenden Teilkomponenten (mit HADES) aufzubauen.

Aufgabe 2.1 Halbaddierer

Ein Halbaddierer liefert die Summe $y = b_1 + b_2$ zweier Bits b_1 und b_2 ohne Berücksichtigung eines vorangegangenen Übertrags gemäß der Wertetabelle (Tab. 1). Dabei stellt $\ddot{u}$ den Übertrag zur nachfolgenden Stelle dar.

b_1	b_2	$\ddot{u}$	y
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

(1)

Realisieren Sie einen Halbaddierer durch Implementation der Booleschen Funktionen $y = f_y(b_1, b_2)$ und $\ddot{u} = f_{\ddot{u}}(b_1, b_2)$ und validieren Sie Ihre Implementation mit HADES.
Erzeugen Sie anschließend mit HADES ein Subdesign (via `Edit/Create Symbol`) für Halbaddierer mit angemesser Schnittstelle, um bei den folgenden Entwurfsprozessen vom internen Aufbau der Schaltung abstrahieren zu können.

Aufgabe 2.2 Volladdierer

Ein Volladdierer liefert die Summe $y = b_1 + b_2$ zweier Bits b_1 und b_2 unter Berücksichtigung eines vorangegangenen Übertrags c gemäß der Wertetabelle (Tab. 2)

b_1	b_2	c	$\ddot{u}$	y
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

(2)

Realisieren Sie einen Volladdierer durch Implementation der boolschen Funktionen $f_y(b_1, b_2, c)$ und $f_{\ddot{u}}(b_1, b_2, c)$. Hinweis: ein Volladdierer läßt sich (mehrstufig) mit Hilfe von zwei Halbaddierern und einem ODER-Gatter aufbauen.
Testen Sie Ihre Implementation des Volladdierers und erstellen Sie ein Symbol für das Subdesign "Volladdierer".

Aufgabe 2.3 4-Bit-Addierer

Implementieren Sie mit Hilfe der zuvor realisierten Subdesigns einen 4-Bit-Addierer (zwei Operanden zu je vier Bit). Überprüfen Sie dessen Richtigkeit durch Simulation mit HADES und erzeugen Sie anschließend für weitere Verwendung ein Symbol für den 4-Bit-Addierer.

Aufgabe 2.4 4-Bit-Subtrahierer

Implementieren Sie einen 4-Bit-Subtrahierer mit HADES.
Bilden Sie zunächst mit Hilfe von vier Invertern das 1-Komplement des Subtrahenden. Anschließend bilden Sie aus dem 1-Komplement das 2-Komplement durch Inkrementieren (Addition von 1). Einen 4-Bit-Inkrementer können Sie mit 1-Bit-Halbaddierern aufbauen.
Durch Addition (unter Verwendung des Addierer-Subdesigns) des 2-Komplements des Subtrahenden zum Minuenden erhalten Sie dann die Differenz.

Aufgabe 2.5 4-Bit-Subtrahierer mit Gültigkeitsprüfung (optional)

Beide Operanden und das Ergebnis können als vorzeichenbehaftete Werte (Wertebereich $-8..7$) in 2-Komplement-Darstellung interpretiert werden. Zeigen Sie, daß der Subtrahierer auch bei dieser Interpretation bereits korrekte Resultate ermittelt.
Erweitern Sie Ihren 4-Bit-Subtrahierer mit einer Schaltung zur Überprüfung der Gültigkeit des Ergebnis.