Prof. Dr. Rolf Drechsler, drechsle@informatik.uni-bremen.de, MZH 3510
Dr. Robert Wille, rwille@informatik.uni-bremen.de, MZH 3485
Philipp Niemann, pniemann@informatik.uni-bremen.de, MZH 3440

1. Übungsblatt zur Vorlesung

Technische Informatik 1

Aufgabe 1 (1+2+2+2 Punkte)

In der Vorlesung wurde die Wahrheitstabelle eines 1-Bit-Addierers angegeben. Für die Funktionen *OR*, *AND* und *NOT* gelten die Tabellen 1(a), 1(b) und 1(c). Als Symbole werden dabei $+$ für *OR*, $\cdot$ für *AND* sowie $\overline{a}$ für *NOT* a verwendet.

(a) Für *OR*

a	b	$a+b$
0	0	0
0	1	1
1	0	1
1	1	1

(b) Für *AND*

a	b	$a \cdot b$
0	0	0
0	1	0
1	0	0
1	1	1

(c) Für *NOT*

a	$\overline{a}$
0	1
1	0

Tabelle 1: Wahrheitstabellen

a) Gib die Wahrheitstabelle der folgenden Funktion an ($\oplus$ bezeichnet die Funktion *XOR*):

$$f_1 = a \oplus b := \overline{a} \cdot b + a \cdot \overline{b}$$

b) Für die *XOR*-Funktion kann man zeigen (ebenso wie für die *OR*- und *AND*-Funktion), dass sie kommutativ und assoziativ ist, in Formeln: $a \oplus b = b \oplus a$ bzw. $(a \oplus b) \oplus c = a \oplus (b \oplus c)$. Mit anderen Worten: die Reihenfolge der Variablen und die Klammerung ist unerheblich für das Ergebnis und kann beliebig verändert werden. Dies darf im Folgenden ohne Beweis verwendet werden.

Zeige zunächst, dass $1 \oplus a = \overline{a}$ gilt. Zeige dann, dass die beiden Booleschen Ausdrücke

$$f_2 = x_2 \oplus (x_1 \oplus x_3) \qquad \text{und}$$

$$g = (x_1 \cdot x_2 + \overline{x_1} \cdot \overline{x_2}) \cdot x_3 + (\overline{x_1} \cdot x_2 + x_1 \cdot \overline{x_2}) \cdot \overline{x_3}$$

äquivalent sind

1. durch Aufstellen einer Wahrheitstabelle,
2. durch algebraische Umformungen (man darf benutzen, dass $\overline{(a \oplus b)} = 1 \oplus a \oplus b = \overline{a} \oplus b$ gilt).

c) Was berechnet die Funktion $f_{n-1} = x_1 \oplus x_2 \oplus \ldots \oplus x_n$? Begründe deine Überlegungen.

d) Dein allwissender Freund Google behauptet, die *XOR*-Funktion könne

1. auch wie folgt definiert werden: $a \oplus b := (a + b) \cdot (\overline{a} + \overline{b})$.
2. ausschließlich durch Verwendung von *AND* und *NOT* ausgedrückt werden (also ohne *OR*).

Ist das korrekt? Begründe deine Antworten!

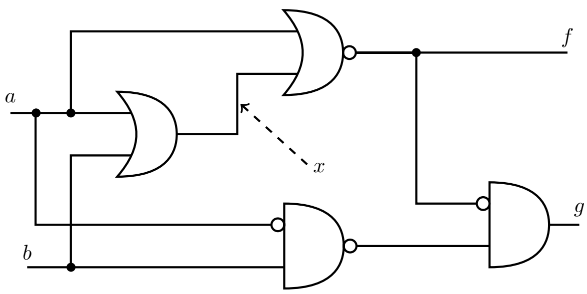

Abbildung 1: Schaltkreis mit „stuck-at-1-fault“

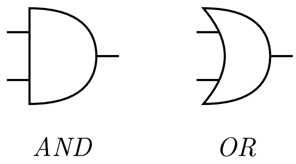

Abbildung 2: Gattertypen

Aufgabe 2 (2+2 Punkte)

a) Wo werden Compiler bzw. Interpreter verwendet? Gib Beispiele an.

b) In welchen Fällen ist die Verwendung eines Compilers bzw. Interpreters vorzuziehen?

Aufgabe 3 (2+2+1 Punkte)

In Abbildung 1 ist ein Schaltkreis dargestellt. Die Gattertypen sind Abbildung 2 zu entnehmen. Jedes Gatter besitzt auf der linken Seite zwei Eingänge und auf der rechten Seite einen Ausgang. Ein- und Ausgänge können mit einem $\circ$ negiert werden. Es wird also eine Funktion dargestellt, die *AND* und *OR* verwendet.

a) Gib die Wahrheitstabelle der durch den Schaltkreis realisierten Funktionen $f(a, b)$ und $g(a, b)$ an. Bestimme hierzu zunächst für jede Funktion den Booleschen Ausdruck, welcher sie repräsentiert.

b) Der Schaltkreis in Abbildung 1 sei nun fehlerhaft, da das Signal an der Stelle x ständig auf dem logischen Wert 1 bleibt (stuck-at-1-fault). Gib die Wahrheitstabellen des fehlerhaften Schaltkreises an, der diesen Fehler berücksichtigt.

c) Wie kann der Fehler festgestellt werden, wenn der Schaltkreis aus a) als Spezifikation für das korrekte Verhalten dient?

Aufgabe 4 (2+2 Punkte)

a) Was leisten die Verifikation und das Testen einer Schaltung?

b) Welches Problem entsteht, wenn eine Schaltung mit n binären Eingängen getestet werden soll?

Abgabetermin: vor Beginn der Vorlesung am 8. Mai 2014