



Klausur Einführung in die Informatik I (Technikorientierung)

05. März 2008

Name:

Matr.-Nr.

Bearbeitungszeit: 120 Minuten

Bewertung

Aufgabe	Punkte	Erreichte Punkte
1	4	
2	6	
3	6	
4	10	
5	9	
6	4	
7	4	
8	5	
9	12	
Summe	60	

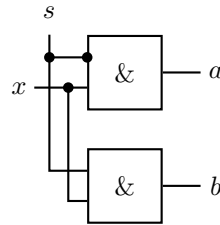
Hinweise:

- Verwenden Sie für die Lösung der Aufgaben **nur** das mit diesem Deckblatt ausgeteilte Papier. Lösungen, die auf anderem Papier geschrieben werden, können **nicht** bewertet werden. Schreiben Sie ihre Lösung auch auf die Rückseiten der Blätter; benötigen Sie für eine Lösung mehr als ein Blatt, finden Sie am Ende der Klausur Leerblätter. Zusätzliches Papier können Sie von den Tutoren bekommen.
- Tragen Sie vor Beginn der eigentlichen Bearbeitungszeit auf **allen** Blättern ihren Namen und ihre Matrikelnummer ein.
- Schreiben Sie deutlich! Unleserliche oder mehrdeutige Lösungen können nicht gewertet werden.
- Schreiben Sie **nicht** mit Bleistift und **nicht** mit rotem oder grünem Stift (das sind die Farben für die Korrektur).
- Als Hilfsmittel sind zwei A4 Blätter, welche beidseitig handschriftlich beschrieben sein dürfen, zugelassen.

Viel Erfolg!

Aufgabe 1 (4 Punkte) Rechneraufbau.

1. (1 Punkt) Um welche Schaltung handelt es sich ?



2. (1 Punkt) Wofür stehen die Ausgangssignale eines Halbaddierers?

3. (1 Punkt) Nennen Sie zwei Unterschiede zwischen einer CISC und einer RISC Architektur.

4. (1 Punkt) Worin besteht der Unterschied zwischen einer präemptiven und nicht-präemptiven Prozessverwaltung?

Aufgabe 2 (6 Punkte) Zahlendarstellungen.

1. (1 Punkt) Wandeln Sie -89_{10} in eine 8-Bit-Zweierkomplementzahl um. Die Rechenschritte müssen erkennbar sein!

2. (2 Punkte)

(a) Addieren Sie die beiden 8-Bit-Zweierkomplementzahlen 01011001_2 und 01010100_2 .

(b) Ist bei der Addition ein Überlauf aufgetreten? Begründen Sie Ihre Antwort mit Hilfe des Überlauf-Bits.

(c) Wandeln Sie das Ergebnis der Addition in eine Dezimalzahl um.



3. (1 Punkt) Stellen Sie die vorzeichenlose Dualzahl 100.110_2 im verkürzten 9-Bit-Gleitkommaformat dar (1-Bit-Vorzeichen, 3-Bit-Exponent, 5-Bit-Mantisse, Exzess=3).

4. (2 Punkte) Multiplizieren Sie die im o.g. 9-Bit-Gleitkommaformat dargestellten Zahlen 011001011 und 001001000 und geben Sie das Ergebnis wieder im 9-Bit-Gleitkommaformat an.

Aufgabe 3 (6 Punkte) ALU.

Gegeben sei eine ALU, die zwei 4-Bit Daten $\vec{a} = (a_3 \dots a_0)$ und $\vec{b} = (b_3 \dots b_0)$ anhand der folgenden ALU-Gleichungen verarbeiten kann.

$$b'_i = (u \wedge b_i) \vee (v \wedge \neg b_i) \tag{1}$$

$$\ddot{u}_i = (a_i \wedge b'_i \vee (\ddot{u}_{i-1} \wedge (a_i \oplus b'_i))) \wedge m \tag{2}$$

$$s_i = a_i \oplus b'_i \oplus \ddot{u}_{i-1} \tag{3}$$

Dabei sei $a \oplus b := (a \wedge \neg b) \vee (\neg a \wedge b)$ die XOR-Verknüpfung. Betrachten Sie den Steuervektor $\vec{st} = (m, u, v) = (1, 1, 1)$.

- (3 Punkte) Vereinfachen Sie die ALU-Gleichungen (1)-(3) für den gegebenen Steuervektor \vec{st} .

- (2 Punkte) Simulieren Sie die ALU mit dem gegebenen Steuervektor \vec{st} für die Eingänge $\vec{a} = (1010)$ und $\vec{b} = (0101)$. Dabei gilt: $\ddot{u}_{-1} = 0$

	3	2	1	0
a	1	0	1	0
b	0	1	0	1
b'				
\ddot{u}				
s				

- (1 Punkt) Welche arithmetische oder logische Funktion führt die ALU für den gegebenen Steuervektor \vec{st} aus?

Aufgabe 4 (10 Punkte) Boolesche Algebra.

Wir betrachten die Boolesche Algebra der Aussagenlogik.

1. (3 Punkte)

(a) Zeigen Sie mit Hilfe einer Wertetabelle, dass die folgenden Booleschen Ausdrücke äquivalent sind:

$$A_1 := (a \wedge b) \vee (a \wedge c) \vee (\neg b \wedge c)$$

$$A_2 := (a \wedge b) \vee (\neg b \wedge c)$$

(b) Wann sind zwei beliebige Boolesche Ausdrücke *nicht* äquivalent? Wie kann man das mit Hilfe einer Wertetabelle feststellen?

2. (5 Punkte) Zeigen Sie mit algebraischer Umformung, dass die folgenden Booleschen Ausdrücke äquivalent sind:

$$A_1 := \neg(\neg a \wedge b) \wedge (a \vee ((a \vee c) \wedge (a \vee \neg c)))$$

$$A_2 := a$$

Der Lösungsweg muss folgenden Bedingungen genügen:

- Es dürfen nur die unten angegebenen Regeln (R1)–(R17) verwendet werden.
- Schreiben Sie bei jeder Umformung über das Gleichheitszeichen die Regel, die Sie verwendet haben. Beispiel:

$$(x \wedge 0) \vee y \stackrel{(R7)}{=} x \vee y$$

- Anwendungen einer der Regeln (R1)–(R4) (Kommutativ und Assoziativgesetze) müssen nicht kenntlich gemacht werden.

(R1) $x \vee (y \vee z) = (x \vee y) \vee z$	(R7) $x \wedge 0 = 0$ und $x \vee 0 = x$	(R13) $x \wedge x = x$
(R2) $x \wedge (y \wedge z) = (x \wedge y) \wedge z$	(R8) $x \vee 1 = 1$ und $x \wedge 1 = x$	(R14) $x \vee x = x$
(R3) $x \vee y = y \vee x$	(R9) $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$	(R15) $\neg\neg x = x$
(R4) $x \wedge y = y \wedge x$	(R10) $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$	(R16) $\neg(x \wedge y) = \neg x \vee \neg y$
(R5) $x \wedge (x \vee y) = x$	(R11) $x \vee \neg x = 1$	(R17) $\neg(x \vee y) = \neg x \wedge \neg y$
(R6) $x \vee (x \wedge y) = x$	(R12) $x \wedge \neg x = 0$	

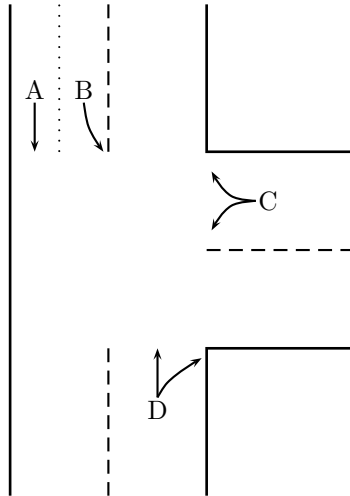
3. (2 Punkte) Wandeln Sie den folgenden Booleschen Ausdruck in eine konjunktive Normalform um. Die Umwandlung muss nachvollziehbar sein.

$$A := (a \vee \neg b) \Rightarrow c$$

Hinweis: Es gilt $x \Rightarrow y = \neg x \vee y$.

Aufgabe 5 (9 Punkte) Schaltungsentwurf.

1. (3 Punkte) Gegeben sei die in der folgenden Abbildung dargestellte Kreuzung. Zur Regelung des Verkehrs werden vier Ampeln A, B, C und D eingesetzt. Jede Ampel besitzt nur zwei Lichtsignale: (i) grün (fahren) und (ii) rot (halten). Das gelbe Lichtsignal wird hier nicht betrachtet. Für jede Ampel A, B, C bzw. D steht genau eine Leitung a, b, c bzw. d zur Verfügung. Auf einer Leitung liegt das Signal 1 genau dann an, wenn die entsprechende Ampel grün ist und andernfalls eine 0.



Ampelsignale (<i>a, b, c</i> und <i>d</i>)	Bedeutung
1	grün
0	rot

- (a) (2 Punkte) Ergänzen Sie zur Prüfung der Ampelanlage folgende Wertetabelle. Dabei liegt am Ausgang *y* eine 1 genau dann an, wenn mindestens zwei sich schneidende Fahrrichtungen mit grün freigegeben werden.

<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>y</i>
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

- (b) (1 Punkt) Lesen Sie aus der Tabelle die ausgezeichnete konjunktive Normalform (AKNF) aus.

2. (4 Punkte) Minimieren Sie die folgende boolesche Funktion mit Hilfe des Quine-McCluskey-Verfahrens.

a	b	c	y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0



-
3. (2 Punkte) Zeichnen Sie die Schaltung zu der Funktion $y = (a \vee b) \wedge (a \vee b)$. Dazu dürfen nur UND- und ODER-Gatter mit zwei Eingängen verwendet werden.

Aufgabe 6 (4 Punkte) Java — Matrizen.

Eine $(n \times n)$ -Matrix $A = (a_{ij})$ heißt M-Matrix, wenn sie folgende Eigenschaften besitzt:

1. Alle ihre Nicht-Diagonalelemente sind negativ, d.h. $a_{ij} < 0$ für alle $i, j \in \{1, 2, \dots, n\}$ mit $i \neq j$.
2. Alle ihre Zeilensummen sind positiv, d.h.

$$\sum_{j=1}^n a_{ij} > 0$$

für alle $i \in \{1, 2, \dots, n\}$.

Folgende Matrix A ist ein Beispiel für eine M-Matrix:

$$A = \begin{pmatrix} 5 & -1 & -2 \\ -3 & 8 & -4 \\ -1 & -1 & 10 \end{pmatrix}$$

Implementieren Sie eine Methode `boolean isMMatrix(double[][] a)`, die `true` genau dann zurückliefert, wenn der spezifizierte Parameter `a` eine M-Matrix ist. Gehen Sie davon aus, dass `a` nicht `null` ist und eine nichtleere quadratische Matrix repräsentiert.

```
boolean isMMatrix(double[][] a) {
```

```
}
```

Aufgabe 7 (4 Punkte) Java — Rekursion.

Betrachten Sie die folgende rekursiv definierte Funktion

$$f : \mathbb{N} \rightarrow \mathbb{N}, \quad f(n) = \begin{cases} 1 & : \text{für } n \leq 1 \\ f(n-1) + 2n - 1 & : \text{für } n > 1 \end{cases}$$

1. (3 Punkte) Implementieren Sie eine rekursive Java-Methode, die $f(n)$ berechnet. Es dürfen keine Schleifen verwendet werden.

Hinweis: Implementieren Sie nur die Methode und verzichten Sie auf eine Klassendefinition.

2. (1 Punkt) Zeichnen Sie den Aufrufbaum von f für $n = 4$.

Aufgabe 8 (5 Punkte) Java — Auswertung von Ausdrücken.

Wie lauten die Ausgaben des folgenden Java-Programms? Schreiben Sie die Ausgaben in die vorgegebenen Bereiche, die als Kommentare gekennzeichnet sind.

```
1  class C {
2
3      int i;
4
5      C() { i = 0; }
6
7      void f(C y) {
8          i += 1;
9          y.i = i + 2;
10     }
11
12     void f(int j) {
13         j *= 2;
14         i = j;
15     }
16
17     public static void main(String[] args) {
18         C x = new C();
19         C y = new C();
20         y.i = x.i++;
21         System.out.println(x.i); //-----
22         System.out.println(y.i); //-----
23
24         x.i = 5;
25         x.f(y);
26         System.out.println(x.i); //-----
27         System.out.println(y.i); //-----
28
29         int j = 7;
30         x.f(j);
31         System.out.println(x.i); //-----
32         System.out.println(j);  //-----
33
34         y.i = 10;
35         x = y;
36         System.out.println(x.i); //-----
37         System.out.println(y.i); //-----
38
39         x.i = 20;
40         System.out.println(x.i); //-----
41         System.out.println(y.i); //-----
42     }
43 }
```

Aufgabe 9 (12 Punkte) Java — Modellierung.

In dieser Aufgaben sollen Sie für eine Bank ein vereinfachtes Java-Programm zur Verwaltung von Bankkonten implementieren.

1. (5 Punkte) Gegeben ist eine leere Klassendefinition der Klasse `Bankkonto`, die ein Bankkonto eines Kunden repräsentiert. Vervollständigen Sie die Klassendefinition gemäß der in den Kommentaren gegebenen Spezifikationen.

```
class Bankkonto {
```

```
/*.....*/
```

```
Attribute (1,5 Punkte)
```

Ein Bankkonto hat eine Kontonummer, ein Guthaben und gehört einer Person, die durch die seine Reisepassnummer repräsentiert wird. Wählen Sie geeignete Typen für die Attribute der Klasse.

```
/*.....*/
```

```
/*.....*/
```

```
Konstruktor (2 Punkte)
```

Der Konstruktor erhält als Parameter eine Kontonummer und eine Reisepassnummer zur Initialisierung der entsprechenden Attribute. Das Guthaben wird mit dem Wert 0 initialisiert.

```
/*.....*/
```



Methode `transaktion(...)` (1,5 Punkte)

Die Methode `transaktion(...)` addiert zum Guthaben einen (positiven oder negativen) Betrag, der als Parameter übergeben wird. Die Transaktion wird nur dann ausgeführt, wenn das neue Guthaben gleich oder größer dem Wert -1000 ist. Andernfalls ist die Transaktion ohne Wirkung. Diese Methode liefert den Wert `true` genau dann zurück, wenn das Guthaben um den spezifizierten Betrag geändert werden konnte.

*****/

} // Ende der Klasse Bankkonto

2. (7 Punkte) Gegeben ist eine leere Klassendefinition der Klasse **Bank**, die eine Bank repräsentiert. Vervollständigen Sie die Klassendefinition gemäß der in den Kommentaren gegebenen Spezifikationen.

```
class Bank {
```

```
/*.....*/
```

Attribute (1 Punkt)

Eine Bank hat eine Bankleitzahl und einen Array von Bankkonten. Wählen Sie geeignete Typen für die Attribute.

```
/*.....*/
```

```
/*.....*/
```

Konstruktor (1,5 Punkte)

Der Konstruktor erhält als Parameter eine Bankleitzahl und die maximale Anzahl der Bankkonten zur Initialisierung der entsprechenden Attribute.

```
/*.....*/
```



```
*****  
Methode neuesKonto(...) (2 Punkte)  
Die Methode neuesKonto(...) erhält als Parameter eine Kontonummer und eine Reisepassnummer. Sie fügt ein  
neues Bankkonto mit der spezifizierten Kontonummer und Reisepassnummer hinzu. Die Methode liefert true  
zurück genau dann, wenn ein neues Bankkonto angelegt werden konnte.  
*****/
```

```
*****  
Methode loescheKonto(...) (2,5 Punkte)  
Die Methode loescheKonto(...) erhält als Parameter eine Kontonummer. Sie löscht das entsprechende Bank-  
konto, sofern es existiert. Die Methode liefert true zurück genau dann, wenn ein bestehendes Bankkonto gelöscht  
werden konnte. Gehen Sie davon aus, dass eine Kontonummer nicht mehrfach existiert.  
*****/
```

```
} // Ende der Klasse Bank
```



Fak. ET/Inform.
Klausur Inftech I
05. März 2008

Name:

Matr.-Nr.



Fak. ET/Inform.
Klausur Inftech I
05. März 2008

Name:

Matr.-Nr.



Fak. ET/Inform.
Klausur Inftech I
05. März 2008

Name:

Matr.-Nr.
