



Klausur Einführung in die Informatik I (Technikorientierung)

28. 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	

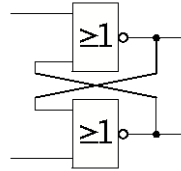
Hinweise:

- 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 aus der Vorlesung bekannte Schaltung handelt es sich ?



2. (1 Punkt) Bitte geben Sie eine Schaltungsskizze für den Halbaddierer an. Verwenden Sie dazu nur die Gatter AND und XOR.

3. (1 Punkt) Nennen Sie zwei der vier in der Vorlesung vorgestellten Funktionseinheiten des Prozessors.

4. (1 Punkt) Nennen Sie drei der fünf Zustände im Lebenszyklus eines Prozesses.

Aufgabe 2 (6 Punkte) Zahlendarstellungen.

1. (1 Punkt) Wandeln Sie -98_{10} in eine 8-Bit-Zweierkomplementzahl um. Die Rechenschritte müssen erkennbar sein!

2. (2 Punkte)

(a) Addieren Sie die beiden 8-Bit-Zweierkomplementzahlen 10001001_2 und 11010100_2 .

(b) Ist bei der Addition ein Unterlauf 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 Dualzahl -010.001_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 000011100 und 011101110 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 $\vec{st} = (m, u, v) = (1, 0, 0)$.

- (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 $\vec{a} = (0111)$ und $\vec{b} = (1011)$. Dabei gilt: $\ddot{u}_{-1} = 1$

	3	2	1	0
a	0	1	1	1
b	1	0	1	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) Überprüfen Sie mit Hilfe einer Wertetabelle, ob die folgenden Booleschen Ausdrücke äquivalent sind:

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

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

(b) Erklären Sie kurz, wie Sie in Aufgabenteil (a) vorgegangen sind, um die beiden Booleschen Ausdrücke auf Äquivalenz zu untersuchen. Begründen Sie Ihre in (a) getroffene Entscheidung.

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

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

$$A_2 := \neg c \wedge b \wedge a$$

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.

$$(R1) \quad x \vee (y \vee z) = (x \vee y) \vee z$$

$$(R2) \quad x \wedge (y \wedge z) = (x \wedge y) \wedge z$$

$$(R3) \quad x \vee y = y \vee x$$

$$(R4) \quad x \wedge y = y \wedge x$$

$$(R5) \quad x \wedge (x \vee y) = x$$

$$(R6) \quad x \vee (x \wedge y) = x$$

$$(R7) \quad x \wedge 0 = 0 \text{ und } x \vee 0 = x$$

$$(R8) \quad x \vee 1 = 1 \text{ und } x \wedge 1 = x$$

$$(R9) \quad x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$$

$$(R10) \quad x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$$

$$(R11) \quad x \vee \neg x = 1$$

$$(R12) \quad x \wedge \neg x = 0$$

$$(R13) \quad x \wedge x = x$$

$$(R14) \quad x \vee x = x$$

$$(R15) \quad \neg\neg x = x$$

$$(R16) \quad \neg(x \wedge y) = \neg x \vee \neg y$$

$$(R17) \quad \neg(x \vee y) = \neg x \wedge \neg y$$

3. (2 Punkte) Wandeln Sie den folgenden Booleschen Ausdruck unter Anwendung algebraischer Umformungen in eine disjunktive Normalform um. Die Umwandlung muss nachvollziehbar sein. (Keine Wertetabelle!)

$$A := (a \vee b) \wedge (c \equiv d)$$

Hinweis: Es gilt $x \equiv y = (x \wedge y) \vee \neg(x \vee y)$.

Aufgabe 5 (9 Punkte) Schaltungsentwurf.

1. (3 Punkte) Es soll ein 5-Bit-Datenpaket (a, b, c, d und y) übertragen werden. Der Empfänger soll erkennen können, ob es zu Übertragungsfehlern gekommen ist. Daher wird der eigentlichen Information im 4-Bit-Code (a, b, c, d) das zusätzliche Prüfbit (y) hinzugefügt. Dieses Prüfbit wird so codiert, dass der 5-Bit-Code (alle Bits a, b, c, d und y) stets eine ungerade Anzahl von Einsen enthält. Wird beim Empfänger eine gerade Anzahl von Einsen ermittelt, handelt es sich um einen Übertragungsfehler.

(a) Bestimmen Sie bitte in nachfolgender Tabelle die Werte des Prüfbits y jeweils so, dass die Datenpakete (a, b, c, d und y) in jeder Zeile eine ungerade Anzahl von Einsen enthalten.

a	b	c	d	y
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 obenstehenden Tabelle die ausgezeichnete disjunktive Normalform (ADNF) 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	1
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 $y = \neg(\neg(a \wedge b) \wedge a)$. Dazu dürfen nur UND- und ODER-Gatter mit zwei Eingängen sowie Negationen verwendet werden.

Aufgabe 6 (4 Punkte) Java — Matrizen.

Betrachten Sie folgende Matrixoperation:

- Zuerst wird das Produkt $b = Ax$ einer $(m \times n)$ -Matrix $A = (a_{ij})$ und eines n -elementigen Vektors $x = (x_j)$ berechnet:

$$b_i = \sum_{j=1}^n a_{ij}x_j, \quad \text{für alle } i \in \{1, 2, \dots, m\}.$$

- Nun wird ein Ergebnisvektor $\mathbf{o} = (o_i)$ zurückgeliefert:

$$o_i = \begin{cases} -1 & : \text{für } b_i \leq 0 \\ +1 & : \text{für } b_i > 0 \end{cases}$$

Hier ein Beispiel für die Berechnung:

$$A = \begin{pmatrix} 1 & 1 & -3 \\ 0 & 2 & 1 \end{pmatrix}, \quad x = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \quad \text{daraus folgt: } b = Ax = \begin{pmatrix} -6 \\ 7 \end{pmatrix} \quad \text{und somit: } o = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

Implementieren Sie eine Methode `double[] calc(double[][] a, double[] x)`, die oben beschriebene Operation auf die Matrix `a` und den Vektor `x` anwendet und den Ergebnisvektor zurückliefert. Gehen Sie davon aus, dass `a` und `x` nicht `null` sind und die Dimensionen zusammenpassen (die Anzahl der Spalten der Matrix entspricht der Anzahl der Elemente des Vektors).

```
double[] calc(double[][] a, double[] x) {
```

```
}
```

Aufgabe 7 (4 Punkte) Java — Rekursion.

Betrachten Sie die folgende rekursiv definierte Funktion:

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

1. (3 Punkte) Implementieren Sie eine rekursive Java-Methode, die $g(n)$ 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 g für $g(3, 5)$.

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[] a;
4
5      C(int n) {
6          a = new int[n];
7          for (int i=0; i<a.length; i++) {
8              a[i] = i;
9          }
10     }
11
12     void f(int[] a) { this.a = a; }
13
14     void f(C y) {
15         a = new int[y.a.length];
16         for (int i=0; i<a.length; i++) {
17             a[i] = y.a[i];
18         }
19     }
20
21     public static void main(String[] args) {
22         C x = new C(2);
23         C y = new C(4);
24         x.a[1] = ++y.a[3];
25         System.out.println(x.a[1]); //-----
26         System.out.println(y.a[3]); //-----
27
28         int[] a = {1, 2, 3};
29         x.a = a;
30         y.f(a);
31         System.out.println(x.a[2]); //-----
32         System.out.println(y.a[0]); //-----
33
34         x.a[2] = 7;
35         System.out.println(a[2]); //-----
36         System.out.println(y.a[2]); //-----
37
38         int[] b = {4, 5};
39         x.a = b;
40         y.f(x);
41         b[1] = 10;
42         System.out.println(x.a[1]); //-----
43         System.out.println(y.a[1]); //-----
44
45         x = y;
46         y.a[1] = 20;
47         System.out.println(x.a[1]); //-----
48         System.out.println(y.a[1]); //-----
49     }
50 }
```

Aufgabe 9 (12 Punkte) Java — Modellierung.

In dieser Aufgaben sollen Sie für eine beliebige Lehrveranstaltung “LV” ein vereinfachtes Java-Programm zur Verwaltung von Studentendaten implementieren.

1. (5 Punkte) Gegeben ist eine leere Klassendefinition der Klasse `Student`, die einen Studenten repräsentiert. Vervollständigen Sie die Klassendefinition gemäß der in den Kommentaren gegebenen Spezifikationen.

```
class Student {
```

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

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

Ein Student hat einen Namen, eine Matrikelnummer und eine aktuelle Gesamtpunktezahl in den bisher erbrachten Übungsleistungen. Wählen Sie geeignete Typen für die Attribute.

```
...../
```

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

```
Konstruktor (2 Punkte)
```

Der Konstruktor erhält als Parameter einen Namen und eine Matrikelnummer zur Initialisierung der entsprechenden Attribute. Die aktuelle Gesamtpunktezahl wird mit dem Wert 0 initialisiert.

```
...../
```



/*****

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

Die Methode `addierePunkte(...)` addiert zur aktuellen Gesamtpunktezahl einen Betrag, der als Parameter übergeben wird. Die Addition wird nur dann ausgeführt, wenn der spezifizierte Betrag größer oder gleich Null ist. Andernfalls ist die Addition ohne Wirkung. Diese Methode liefert den Wert `true` genau dann zurück, wenn die aktuelle Gesamtpunktezahl um den spezifizierten Betrag geändert werden konnte.

*****/

}

2. (7 Punkte) Gegeben ist eine leere Klassendefinition der Klasse LV, die eine Datenbank aller Studenten der Lehrveranstaltung repräsentiert. Vervollständigen Sie die Klassendefinition gemäß der in den Kommentaren gegebenen Spezifikationen.

```
class LV {  
  
    /*****  
    Attribute (1 Punkt)  
    Die Lehrveranstaltung (LV) hat eine LV-Nummer und einen Array von Studenten. Wählen Sie geeignete Typen für die Attribute.  
    *****/
```

```
    /*****  
    Konstruktor (1,5 Punkte)  
    Der Konstruktor erhält als Parameter eine LV-Nummer und die maximale Anzahl der Studenten zur Initialisierung der entsprechenden Attribute.  
    *****/
```




Methodenbeschreibung für `neuerStudent(...)` (2 Punkte)
Die Methode `neuerStudent(...)` erhält als Parameter einen Namen und eine Matrikelnummer. Sie fügt einen neuen Studenten mit dem spezifizierten Namen und Matrikelnummer hinzu. Die Methode liefert `true` zurück genau dann, wenn ein neuer Studenteneintrag angelegt werden konnte.

Methodenbeschreibung für `getPunktezahl(...)` (2,5 Punkte)
Die Methode `getPunktezahl(...)` erhält als Parameter eine Matrikelnummer. Sie liefert die Gesamtpunktezahl des Studenten zurück, sofern der Studenteneintrag existiert. Gehen Sie davon aus, dass eine Matrikelnummer nicht mehrfach existiert.

}



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

Name:
Matr.-Nr.



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

Name:

Matr.-Nr.



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

Name:

Matr.-Nr.
