

14.02.2011

**Lösung zur** Klausur über den Stoff der Vorlesung  
**„Grundlagen der Informatik II“**  
(90 Minuten)

Name: \_\_\_\_\_ Vorname: \_\_\_\_\_

Matr.-Nr.: \_\_\_\_\_ Semester: \_\_\_\_\_ (WS 2010/11)

Ich bestätige, dass ich die folgenden Angaben gelesen und mich von der Vollständigkeit dieses Klausurexemplars überzeugt habe (Seiten 1-18).

\_\_\_\_\_  
Unterschrift des o. g. Klausurteilnehmers  
bzw. der o. g. Klausurteilnehmerin

**Anmerkungen:**

1. Legen Sie bitte Ihren Studierendenausweis bereit.
2. Bitte tragen Sie **Name**, **Vorname** und **Matr.-Nr.** deutlich lesbar ein.
3. Die folgenden **11 Aufgaben** sind vollständig zu bearbeiten.
4. Folgende Hilfsmittel sind zugelassen: **keine**.
5. Täuschungsversuche führen zum Ausschluss von der Klausur.
6. Unleserliche oder mit Bleistift geschriebene Lösungen können von der Klausur bzw. Wertung ausgeschlossen werden.
7. Die Bearbeitungszeit beträgt 90 Minuten.

**Nur für den Prüfer :**

1	2	3	4	5	6	7	8	9	10	11	-	-	-	-	-	gesamt
(9)	(8)	(12)	(13)	(8)	(6)	(13)	(4)	(6)	(6)	(5)						(90)

# Aufgabenübersicht

1) <b>Endliche Automaten</b> (9 Punkte) . . . . .	2
2) <b>Cocke-Younger-Kasami-Algorithmus</b> (8 Punkte) . . . . .	4
3) <b>Turingmaschinen</b> (12 Punkte) . . . . .	5
4) <b>Grammatiken und Pumping-Lemma</b> (13 Punkte) . . . . .	7
5) <b>Berechenbarkeit und Komplexität</b> (8 Punkte) . . . . .	9
6) <b>CMOS</b> (6 Punkte) . . . . .	10
7) <b>Fehlererkennung und Zahlendarstellung</b> (13 Punkte) . . . . .	11
8) <b>Bussysteme</b> (4 Punkte) . . . . .	12
9) <b>Cache-Verfahren</b> (6 Punkte) . . . . .	14
10) <b>Programmiersprachen</b> (6 Punkte) . . . . .	15
11) <b>Betriebssysteme</b> (5 Punkte) . . . . .	17

**Aufgabe 1** **9 Punkte**

2011-H-01

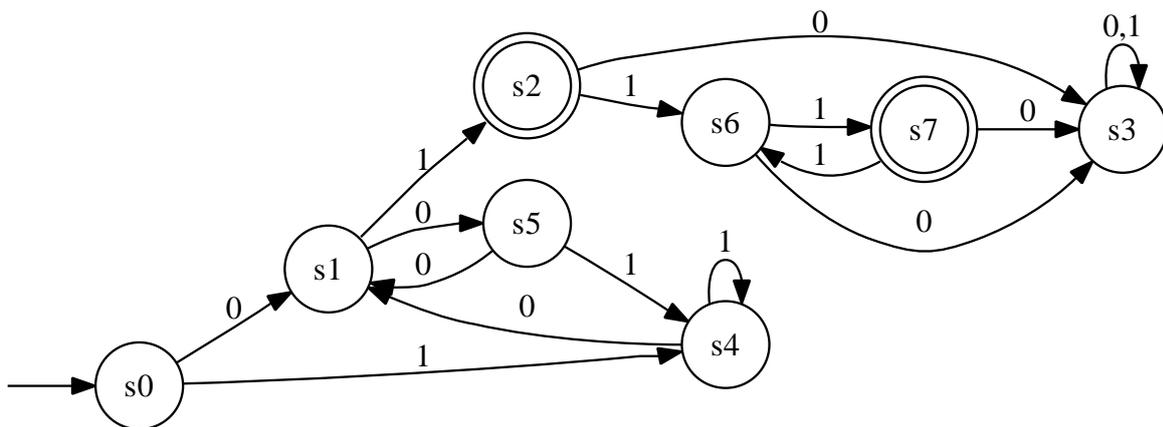
**Endliche Automaten**

/ 9

Gegeben sei der deterministische endliche Automat

$$A = (E, S, \delta, s_0, F) = (\{0, 1\}, \{s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7\}, \delta, s_0, \{s_2, s_7\}).$$

Durch das abgebildete Zustandsdiagramm sei  $\delta$  definiert.

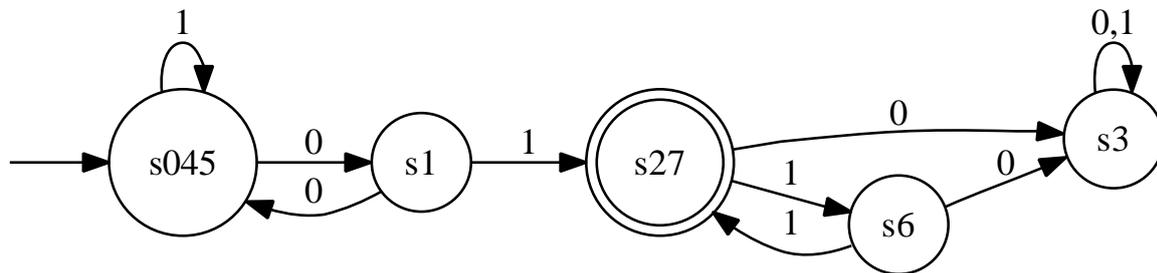


Minimieren Sie  $A$  durch den aus der Vorlesung bekannten Algorithmus und geben Sie den minimierten Automaten  $A' = (E, S', \delta', s'_0, F')$  vollständig an. Nutzen Sie zur Minimierung die vorgegebene Tabelle.

**Lösung:**

	0	1
s0	s1	s4
s1	s5	s2
(s2)	s3	s6
s3	s3	s3
s4	s1	s4
s5	s1	s4
s6	s3	s7
(s7)	s3	s6

s1	X <sub>1</sub>						
s2	X <sub>0</sub>	X <sub>0</sub>					
s3	X <sub>2</sub>	X <sub>1</sub>	X <sub>0</sub>				
s4		X <sub>1</sub>	X <sub>0</sub>	X <sub>2</sub>			
s5		X <sub>1</sub>	X <sub>0</sub>	X <sub>2</sub>			
s6	X <sub>1</sub>	X <sub>3</sub>	X <sub>0</sub>	X <sub>1</sub>	X <sub>1</sub>	X <sub>1</sub>	
s7	X <sub>0</sub>	X <sub>0</sub>		X <sub>0</sub>	X <sub>0</sub>	X <sub>0</sub>	X <sub>0</sub>
	s0	s1	s2	s3	s4	s5	s6



$S' = \{s045, s1, s27, s3, s6\}$ ,  
 $\delta'$  durch Zustandsübergangsdiagramm,  
 $s'_0 = s045$ ,  
 $F' = \{s27\}$ .

**Aufgabe 2**

**8 Punkte**

2011-H-02

**Cocke-Younger-Kasami-Algorithmus**

/ 8
-----

Gegeben sei die Grammatik  $G = (N, T, P, S)$  mit  $N = \{S, A, B, C\}$ ,  $T = \{a, b, c\}$  und

$$\begin{aligned}
 P = \{ & S \rightarrow AC, \\
 & A \rightarrow AB \mid b, \\
 & B \rightarrow BA \mid CB \mid a, \\
 & C \rightarrow CA \mid BB \mid c\}.
 \end{aligned}$$

Überprüfen Sie mithilfe des Algorithmus von Cocke, Younger und Kasami, ob

$$w = abcba \in L(G).$$

Geben Sie zusätzlich zum Ausfüllen der Tabelle explizit an, ob  $w \in L(G)$ .

**Lösung:**

	<i>a</i>	<i>b</i>	<i>c</i>	<i>b</i>	<i>a</i>
<i>m</i> = 1	<i>B</i>	<i>A</i>	<i>C</i>	<i>A</i>	<i>B</i>
<i>m</i> = 2	<i>B</i>	<i>S</i>	<i>C</i>	<i>A</i>	
<i>m</i> = 3	$\emptyset$	<i>S</i>	<i>B, C</i>		
<i>m</i> = 4	$\emptyset$	<i>A, S</i>			
<i>m</i> = 5	<i>B, C</i>				

Da *S* im letzten Feld nicht enthalten ist, liefert der Algorithmus das Ergebnis, dass das angegebene Wort nicht durch die angegebene Grammatik erzeugt werden kann.

**Aufgabe 3**

**12 Punkte**

2011-H-03

**Turingmaschinen**

/ 12
------

Im Folgenden sei die Menge der natürlichen Zahlen definiert als  $\mathbb{N} = \{1, 2, 3, \dots\}$ .

(a) Gegeben sei die Sprache

$$L = \{0^j 10^k \in E^* \mid j, k \in \mathbb{N}, j \geq k\}$$

mit beispielsweise

- 00010, 0010, 0001000, 010, 000000000100  $\in L$
- 1, 0001000000, 10010, 10, 0101010  $\notin L$

Konstruieren Sie eine Turingmaschine  $M = (E, B, S, \delta, s_0, F)$ , die folgende Funktion  $f : E^* \rightarrow E^*$  berechnet:

$$f(w) = \begin{cases} 0^{j-k} & \text{falls } w = 0^j 10^k \in L \\ \text{undefiniert} & \text{falls } w \notin L \end{cases}$$

Bei Eingabe eines Wortes  $w = 0^j 10^k$  soll dieses also in  $w' = 0^{j-k}$  konvertiert werden.

**Hinweise:**

- $M$  soll in einem Endzustand anhalten, falls das Eingabewort  $w$  aus der Sprache  $L$  ist, sonst in einem Nicht-Endzustand.
- $M$  steht zu Beginn über dem linken Zeichen des Eingabewortes  $w$ .
- Nutzen Sie zur Definition von  $M$  die Vorgaben auf der nächsten Seite.
- Geben Sie  $M$  vollständig an.

**Lösung:**

$$E = \{0, 1\}$$

$$B = \{0, 1, \star\}$$

$$S = \{s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_e\}$$

$$F = \{s_e\}$$

$\delta$	0	1	★
$s_0$	$(s_1, ★, R)$		
$s_1$	$(s_1, 0, R)$	$(s_2, 1, R)$	
$s_2$	$(s_2, 0, R)$		$(s_3, ★, L)$
$s_3$	$(s_4, ★, L)$		
$s_4$	$(s_5, 0, L)$	$(s_e, ★, L)$	
$s_5$	$(s_5, 0, L)$	$(s_6, 1, L)$	
$s_6$	$(s_7, 0, L)$		
$s_7$	$(s_7, 0, L)$		$(s_0, ★, R)$
$s_e$			

(b) Beschreiben Sie in kurzen Worten, was bei Ihrer Turingmaschine bei der Eingabe eines Wortes  $w \in L$  passiert. Sie müssen nicht die jeweiligen Zustände angeben, sondern nur die wichtigsten Aspekte der Funktionsweise Ihrer Turingmaschine. **Lösung:**

- $M$  überschreibt erste 0 mit ★.
- $M$  läuft über die restlichen Nullen des Bandes, bis sie auf 1.
- $M$  läuft bis zum Ende des Wortes und überschreibt letzte 0 mit ★.
- $M$  Falls links neben dieser letzten 0 noch eine weitere 0 steht, läuft  $M$  wieder bis zum Beginn der Bandinschrift; Falls keine weitere 0 mehr hinter der 1 steht, wird die 1 mit ★ überschrieben und die Turingmaschine terminiert.

(c) Wie unterscheidet sich das Modell der deterministischen Turingmaschine vom Modell der nichtdeterministischen Turingmaschine bezogen auf die folgenden Eigenschaften (in jeder Zeile ist genau eine Antwort richtig):

**Lösung:**

Sprachmächtigkeit	<input checked="" type="checkbox"/> gleich	<input type="checkbox"/> ungleich	<input type="checkbox"/> unbekannt
Zeitkomplexität (bis auf pol. Faktoren)	<input type="checkbox"/> gleich	<input type="checkbox"/> ungleich	<input checked="" type="checkbox"/> unbekannt

**Aufgabe 4**

**13 Punkte**

2011-H-04

**Grammatiken und Pumping-Lemma**

/ 13
------

Für ein Alphabet  $E$ ,  $w \in E^*$ ,  $a \in E$  bezeichne  $|w|_a$  die Anzahl der  $a$ 's in  $w$ .

Gegeben sei die Sprache

$$L = \{w \in \{a, b, c\}^* \mid \exists n \in \mathbb{N}_0 : |w|_a = n, |w|_b = 2n, |w|_c = 3n\}.$$

Es gilt also bspw.  $\lambda, bbcacc, abbccc, cccabb, abbaccbcbccc \in L$ .

- (a) Geben Sie eine kontextsensitive oder monotone Grammatik  $G = (N, T, P, S)$  an, sodass gilt  $L(G) = L$ . Definieren Sie  $G$  vollständig.

**Hinweis:** Analog zur Vorgehensweise in Vorlesung und Übungen kann die Grammatik zuerst die richtige Anzahl an Zeichen erzeugen und diese dann umsortieren.

**Lösung:**  $N = \{S', S, A, B, C\}, T = \{a, b, c\}$

$$\begin{aligned}
 P = & \{S \rightarrow S' \mid \lambda \\
 & S' \rightarrow S'ABBCCC \mid ABBCCC \\
 & BA \rightarrow AB, \\
 & AB \rightarrow BA, \\
 & CA \rightarrow AC, \\
 & AC \rightarrow CA, \\
 & CB \rightarrow BC, \\
 & BC \rightarrow CB, \\
 & A \rightarrow a, \\
 & B \rightarrow b, \\
 & C \rightarrow c\}
 \end{aligned}$$

- (b) Zeigen Sie mithilfe des Pumping-Lemmas für kontextfreie Sprachen, dass  $L$  nicht kontextfrei ist.

**Lösung:**

Angenommen,  $L$  sei kontextfrei, dann besagt das Pumping-Lemma für kontextfreie Sprachen, es existiert eine Zahl  $k \in \mathbb{N}_0$ , sodass sich jedes Wort  $z \in L$  mit  $|z| \geq k$  zerlegen lässt in  $z = uvwxy$  mit  $u, v, w, x, y \in E^*$  und

- (1)  $|vwx| \leq k$ ,
- (2)  $vx \neq \lambda$ ,

(3) für alle  $i \in \mathbb{N}_0$  gilt  $uv^iwx^iy \in L$ .

Betrachte das Wort  $z = a^k b^{2k} c^{3k}$  mit  $|z| \geq k$ . Wegen (1) können  $v$  und  $x$  maximal zwei verschiedene Zeichen aus  $\{a, b, c\}$  enthalten. Bspw. für  $i = 0$  gilt dann, dass  $uv^iwx^iy \notin L$  (das gilt sogar für alle  $i \neq 1$ ), denn je nach Fall ergeben sich für  $m, n \neq 0$  Wörter

- $a^{k-m} b^{2k} c^{3k}$ ,
- $a^{k-m} b^{2k-n} c^{3k}$ ,
- $a^k b^{2k-m} c^{3k-n}$  oder
- $a^k b^{2k} c^{3k-n}$ ,

die alle nicht in  $L$  enthalten sind. Das stellt mit (2) und (3) einen Widerspruch zu der Annahme dar.

(c) Kreuzen Sie an, zu welchen Sprachklassen der Chomsky-Hierarchie  $L$  nach den Ergebnissen aus (a) und (b) gehört:

- Typ 0
- Typ 1
- Typ 2
- Typ 3

Zu beachten: Typ 0  $\supset$  Typ 1  $\supset$  Typ 2  $\supset$  Typ 3.

**Aufgabe 5**

**8 Punkte**

2011-H-05

**Berechenbarkeit und Komplexität**

/ 8
-----

- (a)  $L_H$  sei die Menge der Kodierungen  $\langle T, w \rangle$  einer Turingmaschine  $T$  mit einer zugehörigen Eingabe  $w$ , sodass  $T$  bei Eingabe von  $w$  hält:

$$L_H = \{ \langle T, w \rangle \mid T \text{ hält auf Eingabe } w \}.$$

Geben Sie für die folgenden aus der Vorlesung bekannten Sprachklassen an, ob  $L_H$  darin liegt oder nicht. (Sie erhalten einen halben Punkt für jedes richtige Kreuz und einen halben Punkt Abzug für jedes falsche Kreuz. Sie können für diese Teilaufgabe nicht weniger als 0 Punkte bekommen.)

$\varnothing(E^*)$	<input checked="" type="checkbox"/> Ja	Nein
$L_0$ bzw. Semientscheidbar	<input checked="" type="checkbox"/> Ja	Nein
Entscheidbar	Ja	<input checked="" type="checkbox"/> Nein
$L_1$ bzw. Kontextsensitiv	Ja	<input checked="" type="checkbox"/> Nein
$L_2$ bzw. Kontextfrei	Ja	<input checked="" type="checkbox"/> Nein
$L_3$ bzw. Regulär	Ja	<input checked="" type="checkbox"/> Nein

- (b) Geben Sie für die folgenden Problem- bzw. Sprachklassen jeweils ein Element (d. h. ein Problem oder eine Sprache) an, das sicher in der Klasse liegt. Beachten Sie, dass keine der Klassen leer ist.

- $NP$ -vollständig:

**Lösung:** SAT, CLIQUE, 3-SAT, TSP, usw.

- $\{Q \mid SAT \leq_{pol} Q\}$ :

**Lösung:** SAT ( $\leq_{pol}$  ist reflexiv), CLIQUE, 3-SAT, TSP, Halteproblem, usw.

- $L_1 \setminus L_2$ :

**Lösung:** Bspw. die Sprache  $L$  von Seite 7, „ $a^n b^n c^n$ “,  $\{ww \mid w \in E^*\}$  für ein beliebiges Alphabet  $E$  (siehe Heimübungsblatt 3), usw.

- $\varnothing(E^*) \setminus L_0$ :

**Lösung:**  $L_{NA}$ , aber NICHT Halteproblem, denn das liegt in  $L_0$ .

**Aufgabe 6**

**6 Punkte**

2011-H-06

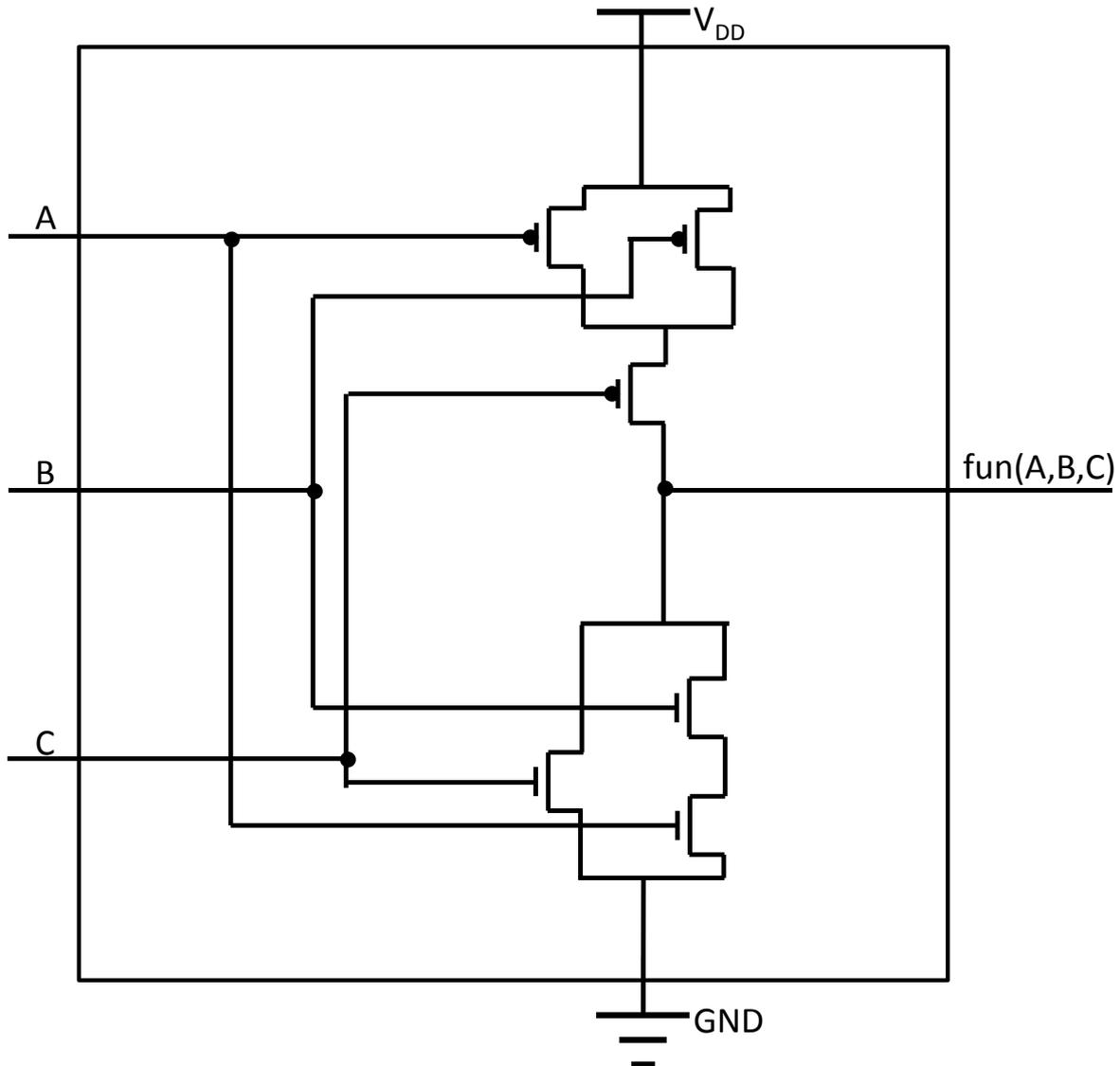
CMOS

/ 6

Gegeben sei die Boolesche Funktion  $fun: \mathbb{B}^3 \rightarrow \mathbb{B}$  mit:

$$fun(A, B, C) = (\neg A \vee \neg B) \wedge \neg C$$

Zeichnen Sie in das vorgegebene Feld eine CMOS-Schaltung für  $fun$ .



**Hinweis:** Versuchen Sie die Funktion direkt in CMOS abzubilden, ohne elementare Bausteine wie NOT, NAND oder NOR zu verwenden.

**Aufgabe 7**

**13 Punkte**

2011-H-07

**Fehlererkennung und Zahlendarstellung**

/ 13
------

- (a) Die Bitstrings  $c(A) = 101101$  und  $c(B) = 010100$  repräsentieren Codewörter für zwei Zeichen 'A' und 'B'.

Wählen Sie als Codewort für ein Zeichen 'C' einen geeigneten Bitstring  $c(C)$  mit  $|c(C)| = 6$ , sodass die komplette Kodierung mindestens 1-Fehler-korrigierbar und mindestens 3-Fehler-erkennbar ist. Geben Sie zunächst den dafür benötigten Hammingabstand an.

**Lösung:**

1-Fehler-korrigierbar:  $k = 1 \leq (h_c - 1)/2 \rightarrow h_c \geq 3$

3-Fehler-erkennbar:  $k = 3 \leq h_c - 1 \rightarrow h_c \geq 4$

$\rightarrow h_c = 4$

derzeitiger  $h_c = 4$

Folgende Lösungen für  $c$  sind möglich: 011011, 000011, 110011, 001010, 111010, 100010.

- (b) Geben Sie die Dezimalzahl  $-49$  als Zahl in 1-Komplementdarstellung mit 8 Bits an.

**Lösung:**

11001110

- (c) Geben Sie die Dezimalzahl 27 in der 8-stelligen Excess- $q$ -Darstellung mit  $q = 127$  an.

**Lösung:**

10011010 (27 wird um 127 verschoben  $\rightarrow 154 = 2^7 + 2^4 + 2^3 + 2^1$ )

- (d) Geben Sie die Dezimalzahl  $-44,75$  in der IEEE 754-Darstellung mit einfacher Genauigkeit (32 Bits) an.

**Lösung:**

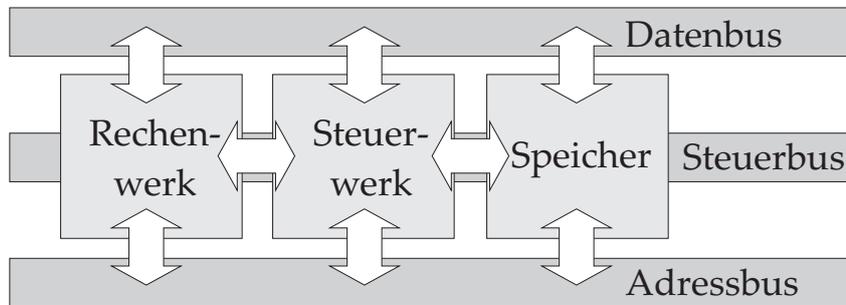
Berechnung:

$44,75 = 2^5 + 2^3 + 2^2 + 2^{-1} + 2^{-2} = 2^5 * (1 + 2^{-2} + 2^{-3} + 2^{-6} + 2^{-7}); 127 + 5 = 132$

$v = 1, c = 10000100, m = 011001100000000000000000$

<b>Aufgabe 8</b>	<b>4 Punkte</b>
<b>2011-H-08</b>	<b>Bussysteme</b>
	/ 4

Gegeben sei die aus der Vorlesung bekannte Busstruktur zwischen Rechen- und Steuerwerk sowie einem Speicher. Jeder Bus habe eine Bandbreite von 32 Bit.



In diesem System sollen zwei 32-Bit-Dualzahlen  $X$  und  $Y$ , die im Speicher an den Adressen  $A_X$  und  $A_Y$  abgelegt sind, miteinander addiert und das Ergebnis  $Z$  an die Adresse  $A_Z$  abgelegt werden.

Tragen Sie auf dem Zeitstrahl ein, welche Daten dazu wann über welche Busse gesendet werden müssen. Der Steuerbus enthält die Information über die Richtung des Datentransfers mit dem Speicher („Lesen“ oder „Schreiben“). Nehmen Sie an, dass ein Datum über einen Bus in einem Takt übertragen wird. Die Rechnung selbst soll ebenfalls einen Takt dauern; die Busbelegung während der Rechnung müssen Sie nicht angeben.

**Lösung:**

Datenbus	Wert $X$	Wert $Y$	Rechnen (keine Daten)	Wert $Z$
Steuerbus	Lesen	Lesen		Schreiben
Adressbus	Adresse $A_X$	Adresse $A_Y$		Adresse $A_Z$
	Takt 1	Takt 2	Takt 3	Takt 4

$t$  →



**Aufgabe 9**

**6 Punkte**

2011-H-09

Cache-Verfahren

/ 6
-----

- (a) Eine CPU, welche über einen 4-zeiligen Directly-Mapped-Cache verfügt, ruft nacheinander Daten von folgenden Adressen auf:

101, 102, 103, 106, 105, 106, 105, 102.

Geben Sie für jede der Zeilen im Cache an, welche Information dort nach allen Aufrufen im Tagfeld steht.

**Lösung.**

Der Cache enthält 4 Blöcke und der Block mit der Adresse  $z$  gelangt an die Position  $z \bmod 4$  im Cache.

Zeilennummer	Tagfeld
0	(leer)
1	105 <i>div</i> 4
2	102 <i>div</i> 4
3	103 <i>div</i> 4

- (b) Eine andere CPU benutzt einen 4-zeiligen Assoziativ-Cache, welcher Daten an die erste freie Stelle schreibt bzw., falls notwendig, nach dem Least-Recently-Used-Prinzip alte Daten aus dem Cache löscht. Wieder werden folgende Adressen aufgerufen:

101, 102, 103, 106, 105, 106, 105, 102.

Geben Sie für jede der Zeilen im Cache an, welche Information dort nach allen Aufrufen im Tagfeld steht.

**Lösung.**

Zu Beginn werden alle 4 Blöcke nacheinander gefüllt, dann wird der am längsten nicht benutzte Block überschrieben.

Zeilennummer	Tagfeld
0	105
1	102
2	103
3	106

- (c) Gegeben sei ein Multicore-System, bei dem mehrere Prozessoren mit jeweils einem eigenen Cache auf einen gemeinsamen Hauptspeicher zugreifen. Welche der Cache-Strategien write-back bzw. write-through eignet sich besser für dieses System? Begründen Sie Ihre Antwort.

**Lösung:** write-through, denn der Hauptspeicher muss konsistent gehalten werden. Sonst könnte ein Prozessor-Kern Änderungen am Speicher durch einen anderen Prozessor nicht mitbekommen.

**Aufgabe 10**

**6 Punkte**

2011-H-10

**Programmiersprachen**

/ 6

(a) Beschreiben Sie für die folgenden Kategorien problemorientierter Programmiersprachen die wesentlichen Eigenschaften der zugehörigen Programmiersprachen.

- Imperative Programmiersprachen (I):

**Lösung:**

Programme mit algorithmischem Charakter: genaue Festlegung der Folge von Befehlen an den Computer.

Deutliche Widerspiegelung der Architektur des von-Neumann-Rechners.

- Funktionale oder applikative Programmiersprachen (F):

**Lösung:**

Programme sind Funktionen von Eingabemengen in Ausgabemengen, zu lösende Probleme werden funktional beschrieben.

funktionale und applikative Programme legen nicht die zeitliche Reihenfolge auszuführender Operationen fest, sondern nur Reihenfolgen aufgrund logischer Abhängigkeiten.

beiden gemeinsam: funktionale Beschreibung, d. h. Beschreibung von Abhängigkeiten zwischen Dingen und Vorgängen der realen Welt.

- Prädikative oder logische Programmiersprachen (P):

**Lösung:**

Programme als Mengen von Fakten (gültige Prädikate) und Regeln (wie man aus Fakten neue Fakten gewinnt).

Man formuliert „eigenes Wissen“ über das Problem; der Computer versucht, durch dieses Wissen die Lösung des Problems herzuleiten.

Geeignet für Probleme, die sich einfach in einem logischen Kalkül (siehe Info I) formulieren lassen.

- Objektorientierte Programmiersprachen (O):

**Lösung:**

Zentraler Gegenstand der Programmiersprache: (aktive) Objekte

Objekte haben zugeordnete Eigenschaften (Attribute) und Methoden (Prozeduren) und werden zu Klassen gleichartiger Objekte zusammengefasst.

wesentliche Konzepte:

Geheimnisprinzip: Zugriff auf Objekte nur über „Botschaften“ (Aufruf zugeordneter Methoden)

Vererbungsprinzip: Eigenschaften können von einer Klasse an Unterklassen vererbt werden.

Dynamische Bindung: Zuordnung von Methoden zu Objekten erfolgt dynamisch zur Laufzeit.

- (b) Geben Sie für die folgenden Programmiersprachen an, zu welcher der obigen Kategorien (I, F, P oder O) sie gehören. Pro Zeile muss nur eine Kategorie angegeben werden.

Programmiersprachen	Kategorie (I, F, P, O)
SMALLTALK-80, C++, Java	O
LISP, LOGO, Gofer	F
PROLOG	P
ADA, BASIC, C	I

<b>Aufgabe 11</b>	<b>5 Punkte</b>
<b>2011-H-11</b>	<b>Betriebssysteme</b>
	/ 5

(a) Nennen Sie zwei typische Dienste von Betriebssystemen.

**Lösung:**

Prozessorverwaltung, Hauptspeicherverwaltung, Programmausführung, Zugriff auf E/A-Geräte, Netzzugriff, Zugriff auf Dateien, Systemzugriff, Buchführung

(b) Ordnen Sie in der folgenden Tabelle den verschiedenen Betriebsarten eines Betriebssystems die entsprechenden Hauptaufgaben zu, indem Sie bei der Hauptaufgabe die entsprechende Nummer der Betriebsart in ( ) eintragen.

**Lösung:**

Betriebsart	Hauptaufgabe
(1) Stapelbetrieb	(5) effiziente Abwicklung entfernter Prozeduraufrufe
(2) Multiprogrammbetrieb	(4) größtmögliche/zeitgerechte Verfügbarkeit der Betriebsmittel
(3) Dialogbetrieb	(1) Maximierung des Durchsatzes
(4) Echtzeitbetrieb	(3) Minimierung der Antwortzeit
(5) Client/Server-Betrieb	(2) Zuteilung von Betriebsmitteln an Prozesse (Aktivierung, Blockierung, gegenseitige Isolation)

(c) Für fünf bereite Prozesse  $P_1, \dots, P_5$  seien folgende CPU-Zeiten zu erwarten:

- $Z(P_1) = 13 \text{ ms}$
- $Z(P_2) = 5 \text{ ms}$
- $Z(P_3) = 20 \text{ ms}$
- $Z(P_4) = 17 \text{ ms}$
- $Z(P_5) = 3 \text{ ms}$

Teilen Sie die Rechenzeit gemäß dem Round-Robin-Verfahren zu. Die Zeitscheibe sei dabei in feste Zeitspannen der Länge  $Z = 10 \text{ ms}$  unterteilt. Veranschaulichen Sie Ihr Ergebnis auf dem unten gegebenen Zeitstrahl.

**Lösung:**

