

19.07.2010

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

Name: _____ Vorname: _____

Matr.-Nr.: _____ Semester: _____ (WS 2009/10)

Ich bestätige, dass ich die folgenden Angaben gelesen und mich von der Vollständigkeit dieses Klausurexemplars überzeugt habe (Seiten 1-16).

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
(10)	(8)	(10)	(10)	(7)	(5)	(6)	(8)	(11)	(9)	(6)						(90)

Aufgabenübersicht

1) Pumping-Lemma, kontextfreie Grammatiken (10 Punkte) . . .	2
2) Reguläre Sprachen] (8 Punkte)	3
3) Turingmaschinen (10 Punkte)	4
4) Komplexitätstheorie (10 Punkte)	5
5) Binary-Decision-Diagram (7 Punkte)	7
6) CMOS (5 Punkte)	9
7) Huffman-Kodierung (6 Punkte)	10
8) Zahlendarstellung (8 Punkte)	11
9) Von-Neumann-Rechner, Pipelining (11 Punkte)	13
10) Assembler (9 Punkte)	15
11) Betriebssysteme und Dateioorganisation (6 Punkte)	16

Aufgabe 1**10 Punkte****2010-N-01****Pumping-Lemma, kontextfreie Grammatiken**

/ 10

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

Ein wohlgeformter Klammerausdruck ist ein Wort der Sprache

$$L = \{w \in \{(,)\}^* \mid |w|_((= |w|_{)) \text{ und } w = uv \Rightarrow |u|_{((} \geq |u|_{))}\}.$$

Ein Wort $w \in L$ darf also jede Kombination aus öffnenden und schließenden Klammern sein, bei der die Gesamtanzahl von öffnenden Klammern der Gesamtanzahl von schließenden Klammern entspricht und bei der in jedem Präfix von w höchstens so viele schließende wie öffnende Klammern vorkommen.

Beispiele für wohlgeformte Klammerausdrücke sind: $\lambda, (()), ()(), ()((())()) \in L$.

- (a) Zeigen Sie mithilfe des Pumping-Lemmas für reguläre Sprachen, dass L nicht regulär ist.

- (b) Zeigen Sie durch Angabe einer kontextfreien Grammatik G mit $L(G) = L$, dass L kontextfrei ist. Definieren Sie die Grammatik vollständig.

Aufgabe 2**8 Punkte****2010-N-02****Reguläre Sprachen]**

/ 8

Wie in Aufgabe 1 sei L die Sprache der wohlgeformten Klammerausdrücke:

$$L = \{w \in \{(,)\}^* \mid |w|_{(} = |w|_{)} \text{ und } w = uv \Rightarrow |u|_{(} \geq |u|_{)}\}.$$

Für $n \in \mathbb{N}$ bezeichne L_n die wohlgeformten Klammerausdrücke mit einer Länge von höchstens n :

$$L_n = \{w \in L \mid |w| \leq n\}.$$

Es gilt bspw. $L_2 = \{(), \lambda\}$.

(a) Geben Sie L_4 an.

(b) Geben Sie einen deterministischen endlichen Automaten A an mit $L(A) = L_4$. Definieren Sie den Automaten vollständig.

Aufgabe 4**10 Punkte****2010-N-04****Komplexitätstheorie**

--

 / 10

(a) Geben Sie für die folgenden Komplexitätsklassen jeweils ein aus der Vorlesung bekanntes Problem an, das sicher in der jeweiligen Klasse liegt, oder begründen Sie, warum Sie keines angeben können.

(1) P

(2) NP

(3) $P \cap NP$ -vollständig

(4) $\{R \mid \forall Q \in NP: Q \leq_{pol} R\}$

(5) $DTime(n^3) \setminus P$

(b) Für zwei Probleme Q, R gelte: $Q \leq_{pol} R$.

(1) Welche Aussage über die Schwierigkeit der Probleme Q, R wird hier getroffen?

(2) Was bedeutet es für Q , wenn $R \in P$ gilt?

(3) Was bedeutet es für R , wenn $Q \in NP$ gilt?

Aufgabe 5**7 Punkte****2010-N-05****Binary-Decision-Diagram**

/ 7

Gegeben sei folgende Boolesche Funktion $f : \mathbb{B}^3 \rightarrow \mathbb{B}$ mit

$$f(a, b, c) = (b + a) \cdot (c' + a')$$

- (a) Stellen Sie f in der nachstehenden Tabelle dar. Geben Sie dafür zunächst die Komponenten $(b + a)$ und $(c' + a')$ an.

a	b	c	$(b + a)$	$(c' + a')$	$f(a, b, c)$
1	1	1			
1	1	0			
1	0	1			
1	0	0			
0	1	1			
0	1	0			
0	0	1			
0	0	0			

- (b) Erzeugen Sie das Binary-Decision-Diagram (BDD) zu f mit der Variablenreihenfolge a - b - c. Benutzen Sie hierzu den Platz auf der kommenden Seite.

BDD:

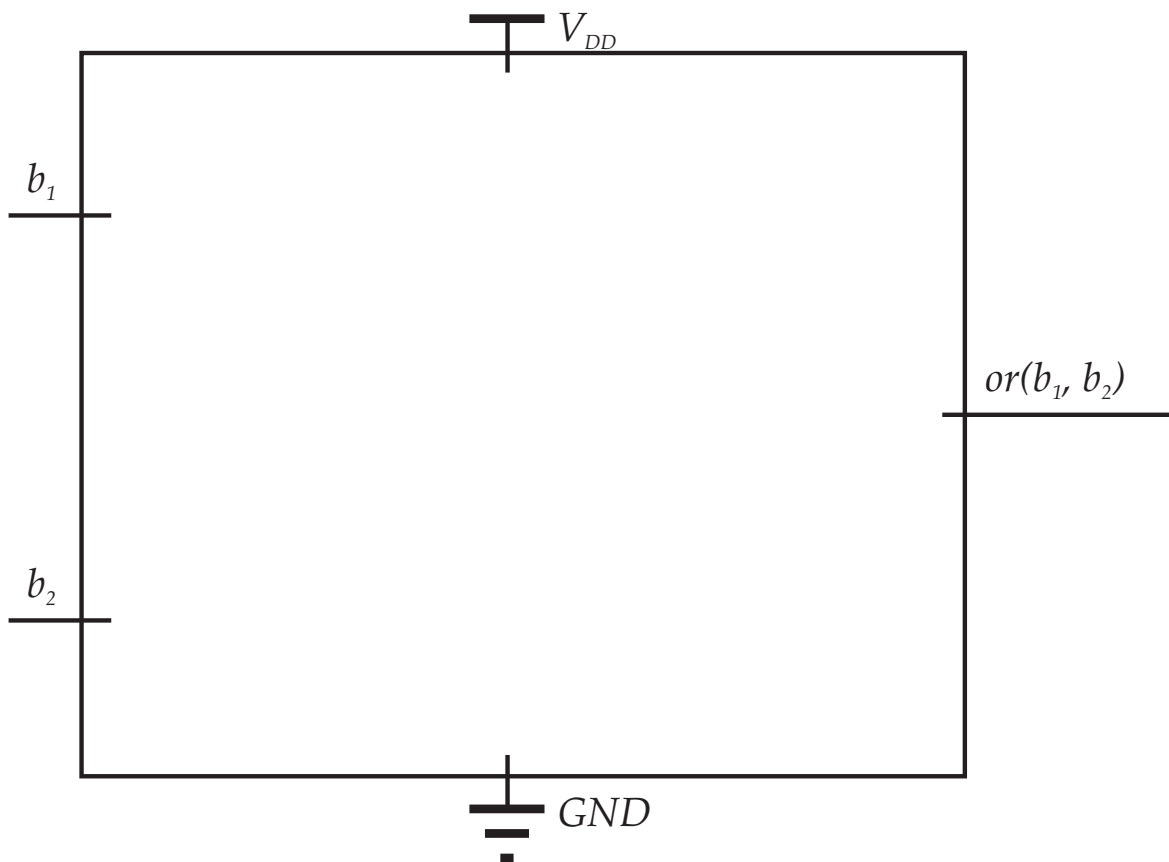
Aufgabe 6**5 Punkte****2010-N-06****CMOS**

/ 5

Die Boolesche Funktion $or: \mathbb{B}^2 \rightarrow \mathbb{B}$ sei für $b_1, b_2 \in \mathbb{B}$ definiert als

$$or(b_1, b_2) = b_1 \vee b_2.$$

(a) Zeichnen Sie in das vorgegebene Feld eine CMOS-Schaltung, die or implementiert.



Hinweis: Überlegen Sie zunächst, welche CMOS-Schaltungen Sie bereits kennen und wie man or durch diese (eventuell durch Verknüpfung mehrerer) darstellen kann.

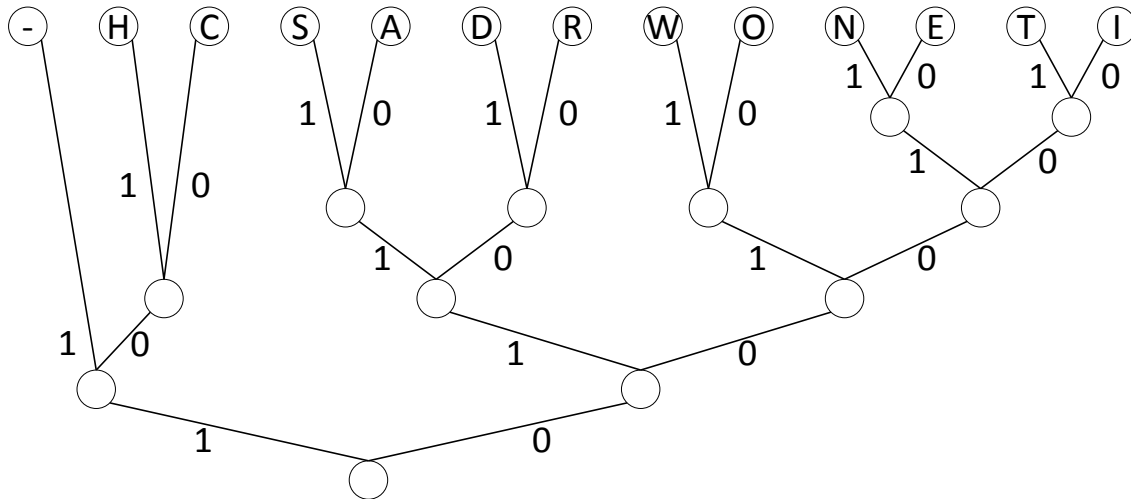
Aufgabe 7 **6 Punkte**

2010-N-07

Huffman-Kodierung

/ 6

Gegeben sei folgender Huffman-Baum.



(a) Dekodieren Sie anhand des obigen Huffman-Baums den folgenden String.

0001100000100101000011101111001010011000100100

(b) Die folgende Tabelle gibt die relative Häufigkeit einzelner Zeichen in einem zu kodierenden Text an. Diese Verteilung wurde auch dem obigen Huffman-Baum zugrunde gelegt.

-	H	C	S	A	D	R	W	O	N	E	T	I
0,16	0,12	0,12	0,08	0,08	0,08	0,08	0,08	0,08	0,04	0,04	0,04	0,04

Bisher seien die einzelnen Zeichen jeweils durch 4 Bits kodiert worden. Wie groß ist die erwartete relative Ersparnis an Speicherplatz bei Verwendung des angegebenen Huffman-Codes?

Aufgabe 8**8 Punkte****2010-N-08****Zahlendarstellung**

--

 / 8

Gegeben seien die folgenden beiden Binärsequenzen:

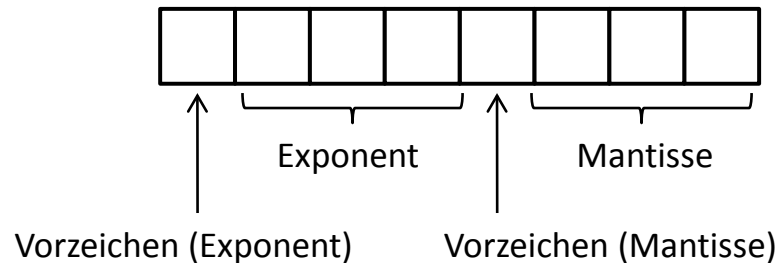
$$a = 10110101, b = 01111001$$

- (a) Berechnen Sie $a + b$, wobei a, b als Zahlen in Dualdarstellung zu interpretieren sind. Geben Sie das Ergebnis sowohl in Dualdarstellung, als auch in Dezimaldarstellung an. Bei der Dezimaldarstellung reicht die Angabe der Summe der Zweierpotenzen.

- (b) Wieder sei die Binärsequenz a als Zahl in Dualdarstellung zu interpretieren. Bilden Sie das 2-Komplement von a .

- (c) Betrachten Sie die folgende Darstellung einer 8-Bit-Gleitkommazahl. Der Exponent und die normierte Mantisse ohne führende Eins sind in Betrag-Vorzeichen-Darstellung gespeichert. Die Basis für den Exponenten ist 2.

Hinweis: Beachten Sie, dass diese Darstellung einer Gleitkommazahl nicht der IEEE-754-Kodierung entspricht.



Nehmen Sie für a und b diese Darstellung einer 8-Bit-Gleitkommazahl an. Geben Sie den Dezimalwert für a und b an. Auch hier reicht die Angabe der Summe der Zweierpotenzen.

Aufgabe 9 **11 Punkte**

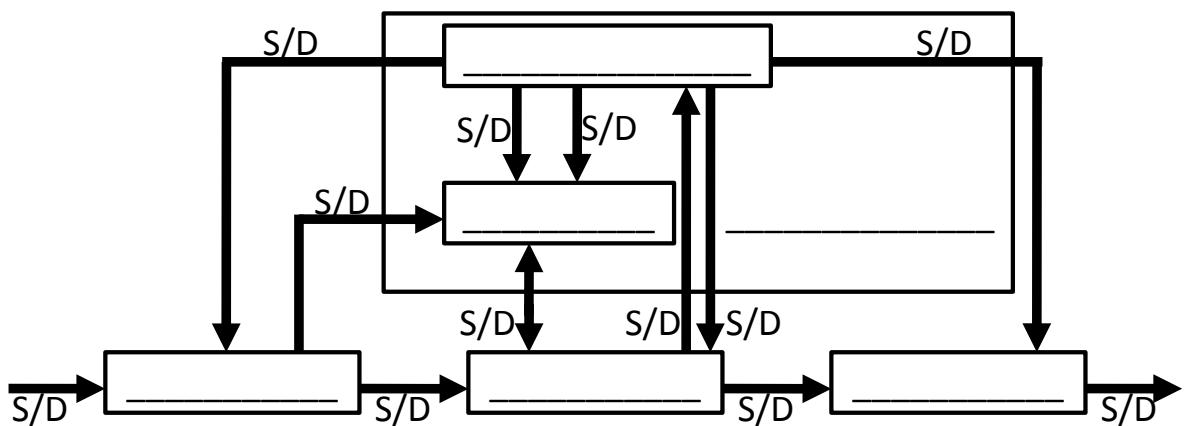
2010-N-09

Von-Neumann-Rechner, Pipelining

/ 11

In der Vorlesung wurde das Von-Neumann-Rechnermodell vorgestellt.

- (a) Benennen Sie die einzelnen Komponenten des Von-Neumann-Rechners in der nachstehenden Skizze. Kennzeichnen Sie zudem, bei welchen der angegebenen Flüsse (Pfeile) es sich um Datenfluss (D) und bei welchen um Steuerfluss (S) handelt, indem Sie an den Pfeilen die jeweils falsche Bezeichnung streichen.



- (b) Wo liegt eine wesentliche Schwachstelle der Von-Neumann-Architektur? Beschreiben Sie diese kurz.
- (c) Von-Neumann-Rechner gehören zur Klasse des SISD-Architektur. Wofür steht die Abkürzung SISD und was bedeutet diese?

- (d) Durch die Verwendung des Pipelining können verschiedene Befehle parallel ausgeführt werden.

Veranschaulichen Sie in den nachstehenden Diagrammen die Abarbeitung dreier Befehle mit und ohne Pipelining graphisch. Nehmen Sie dabei an, dass jeder der Befehle 1, 2, 3 in die vier Teilaufgaben $\boxed{F_i}$, $\boxed{D_i}$, $\boxed{O_i}$, $\boxed{E_i}$, $i \in \{1, 2, 3\}$ zerlegt werden kann.

Abarbeitung ohne Pipelining



Abarbeitung mit Pipelining



- (e) Welches Problem kann bei einer parallelen Abarbeitung von Befehlen wie dem Pipelining auftreten?

Aufgabe 10**9 Punkte****2010-N-10****Assembler**

--

 / 9

Die Befehle der in der Vorlesung vorgestellten Assembler-Sprache sind folgendermaßen aufgebaut, wobei 'Q' für Quelle steht und 'Z' für Ziel:

OpCode Q1, (Q2,) Z

Für unmittelbare Adressierung wird das Präfix '#' verwendet. Ein bedingter Sprungbefehl ist *JNZ* (JumpNotZero), der den Befehlszähler genau dann zu Label *L* springen lässt, falls $Q \neq 0$. Die Notation des Sprungbefehls ist:

JNZ Q L

Gegeben sei das folgende Assemblerprogramm.

1. STORE #1, R2
2. L1: MULTIPLY R2, R1, R2
3. SUBTRACT R1, #1, R1
4. JNZ R1, L1
5. STORE R2, R1

(a) Zu Beginn stehe in Register *R1* ein Wert $n > 0, n \in \mathbb{N}$. Wie hängt der Endwert $R1^E$ des Registers *R1* mit seinem Anfangswert $R1^A = n$ zusammen, welche Funktion berechnet das Programm also bezogen auf *R1*?

(b) Was geschieht, wenn zu Beginn der Wert $R1^A = 0$ in *R1* steht?

(c) Welche Funktion berechnet das Programm bezogen auf *R1*, wenn man zwischen Zeile 2 und Zeile 3 zusätzlich noch den Befehl

ADD R2, R2, R2

einfügt?

Aufgabe 11**6 Punkte****2010-N-11****Betriebssysteme und Dateorganisation**

/ 6

Kreuzen Sie an, ob die folgenden Aussagen wahr oder falsch sind. Für jede richtige Antwort erhalten Sie einen Punkt, für falsche Kreuze wird Ihnen jeweils ein Punkt abgezogen. Sie können nicht weniger als 0 Punkte in dieser Aufgabe erzielen.

- | | wahr | falsch |
|--|-----------------------|-----------------------|
| Das Round-Robin-Zeitscheibenverfahren berücksichtigt von Prozessen einzuhaltende Fristen. | <input type="radio"/> | <input type="radio"/> |
| Threads innerhalb desselben Prozesses nutzen einen gemeinsamen Speicherbereich und können daher effizient kommunizieren. | <input type="radio"/> | <input type="radio"/> |
| Ein Prozess kann vom Zustand „bereit“ direkt in den Zustand „blockiert“ wechseln. | <input type="radio"/> | <input type="radio"/> |
| Das Zuteilungsverfahren First-Come-First-Serve (FCFS) ist ein non-preemptive Verfahren. | <input type="radio"/> | <input type="radio"/> |
| Bei der Dateorganisation ist die Hash-Organisation eine Art der Primärorganisation. | <input type="radio"/> | <input type="radio"/> |
| Eine Hashfunktion ist immer bijektiv, damit aus dem Primärschlüssel eine eindeutige Adresse berechnet werden kann. | <input type="radio"/> | <input type="radio"/> |