



Klausur Technische Grundlagen der Informatik
Studiengang Medieninformatik
H. Linnemann

Freitag, 03. Februar 2012, 12.00 Uhr, Raum D102/H2

- Zugelassene Hilfsmittel: Keine.
- Versehen Sie bitte jedes Lösungsblatt mit Ihrem Namen und mit einer fortlaufenden Seitennummer.
- Falls Teillösungen über mehrere Seiten verteilt sind, versehen Sie diese bitte mit entsprechenden Querverweisen. Nicht gekennzeichnete oder nicht eindeutig zugeordnete Lösungsfragmente werden nicht gewertet!
- Reklamationen der Korrektur und Bewertung nur bei Klausurrückgabe!

Name:

Vorname:

Matrikel-Nr.

Unterschrift:

Aushändigen der korrigierten Klausur (zutreffendes bitte ankreuzen):

- Nur an mich persönlich oder an Kommilitonen/innen mit schriftlicher Vollmacht
- An Frau / Herrn:
- An alle, die danach fragen

Dritter (letzter zulässiger) Versuch oder Prüfungsfrist läuft in diesem Semester ab:

- Nein. ✓
- Ja.

Frage	Max. Punkte	Erreichte Punkte
Frage 1	6	5
Frage 2	8	8
Frage 3	14	14
Frage 4	15	13
Frage 5	8	8
Frage 6	9	9
Frage 7	17	17
Frage 8	12	12
Frage 9	11	2
SUMME	100	88

NOTE: 1,7

- 1.) Wandeln Sie die Oktalzahl 635342,413 in das Dual- und das Hexadezimalsystem!
- 2.) Es sind zwei Hexadezimalzahlen gegeben: $a = 7A3C_{16}$, $b = 466A_{16}$.
 - a) Berechnen Sie im hexadezimalen Zahlensystem $a + b$ und $a - b$.
 - b) Führen Sie die Subtraktion zusätzlich im 16er Komplement aus.
- 3.) Stellen Sie die Dezimalzahl 75,875 dual im Fließkommaformat dar. Verwenden Sie das 32-Bit-Fließkommaformat mit 1 Bit Vorzeichen, 8 Bit Exponent mit Offset 127 und 23 Bit Fraktion.
- 4.) Gegeben ist folgende Funktionstabelle eines BCD zu 7-Segment-Dekoders:

BCD-Code (Eingangsvariable)				7-Segment-Code (Ausgangsvariable)						
D	C	B	A	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1
1	0	1	0	X	X	X	X	X	X	X
1	0	1	1	X	X	X	X	X	X	X
1	1	0	0	X	X	X	X	X	X	X
1	1	0	1	X	X	X	X	X	X	X
1	1	1	0	X	X	X	X	X	X	X
1	1	1	1	X	X	X	X	X	X	X

} Pseudo-tetraden (Don't Care)

Ermitteln Sie **nur für die Ausgangsvariable a!** Wählen Sie eine andere Variable, werden Null Punkte vergeben!

- a) die disjunktive Normalform.
- b) das zugehörige KV-Diagramm und die minimierte Funktion. Berücksichtigen Sie dabei die Don't Care und verwenden Sie bitte das folgende in Ihr Lösungsblatt zu übertragende Schema:

	CD			
AB	00	01	11	10
00				
01				
11				
10				

- c) Entwickeln sie für die minimierte Funktion das Schaltnetz.

- 5.) Ein zentraler Begriff der Informatik ist die sog. Von Neumann Architektur.
- Nennen Sie vier wesentliche Eigenschaften des von Neumann-Rechners!
 - Was versteht man unter dem „von Neumann’schen Flaschenhals“?
 - Wie kann man ihn vermeiden und wie heißt diese Rechnerarchitektur?
- 6.) Interrupts von Intelprozessoren können in drei Klassen eingeteilt werden. Nennen Sie die drei Klassen, die Art ihrer Auslösung und geben Sie jeweils ein Anwendungsbeispiel.
- 7.) Aktuelle Grafikkarten verfügen über 3D-Beschleuniger zur Unterstützung des Rendering.
- Was versteht man unter Rendering?
 - Nennen Sie die drei Stufen des Ebenenmodells der so genannten Rendering-Pipeline für 3D-Darstellungen und charakterisieren Sie die in den einzelnen Stufen vorgenommenen Arbeiten.
- 8.) Nach dem Umfang und der Komplexität des Befehlsvorrats unterscheidet man zwei Grundtypen von Prozessoren.
- Nennen Sie die beiden Grundtypen (sowohl die Abkürzung als auch die „Langform“).
 - Stellen Sie die wesentlichen Eigenschaften sowie Vor- und Nachteile der beiden Grundtypen gegenüber.

- 9.) Gegeben ist der folgende Ausschnitt eines Assemblerprogramms (in einem durch eine Null begrenzten String werden enthaltene Kleinbuchstaben in Grossbuchstaben umgewandelt; die ASCII-Codes sind: 'A' = 41H, 'a' = 61H, 'B' = 42H, 'b' = 62H, 'c' = 63H, 'C' = 43H, 'z' = 7AH, '\$' = 24H).

Befehl	Register							
	SI	DL	SI	DL	SI	DL	SI	DL
	1. Durch- lauf	2. Durch- lauf	3. Durch- lauf	4. Durch- lauf	3. Durch- lauf	4. Durch- lauf	3. Durch- lauf	4. Durch- lauf
. . .								
.DATA								
OTTO DW 0,0 ;Irgend etwas								
STRING DB 'aCb',0 ;Der OFFSET von String ist 4!								
.CODE								
MOV AX,@DATA								
MOV DS,AX								
MOV SI, OFFSET STRING	04H							
AUSGABE:								
MOV DL, [SI]		61H		42H		43H		44H
CMP DL, 0								
JE GOON								
CMP DL, 61H 'a'								
JL NEXTBUCH								
CMP DL, 7AH 'z'								
JG NEXTBUCH								
SUB DL, 20H				44H				
MOV [SI], DL	41H							
NEXTBUCH:								
INC SI	02H	42H		43H	44H	45H		
JMP AUSGABE								
GOON:								
DEC SI								
. . .								

2/12

Tragen Sie die NACH der Ausführung der einzelnen Befehle in den oben angegebenen Registern vorhandenen Werte in die Tabelle ein.

- 9.) Gegeben ist der folgende Ausschnitt eines Assemblerprogramms (in einem durch eine Null begrenzten String werden enthaltene Kleinbuchstaben in Grossbuchstaben umgewandelt; die ASCII-Codes sind: 'A' = 41H, 'a' = 61H, 'B' = 42H, 'b' = 62H, 'c' = 63H, 'C' = 43H, 'z' = 7AH, '\$' = 24H).

Befehl	Register							
	SI	DL	SI	DL	SI	DL	SI	DL
	1. Durch- lauf	2. Durch- lauf	3. Durch- lauf	4. Durch- lauf	3. Durch- lauf	4. Durch- lauf	3. Durch- lauf	4. Durch- lauf
. . .								
.DATA								
OTTO DW 0,0 ;Irgend etwas								
STRING DB 'aCb',0 ;Der OFFSET von String ist 4!								
.CODE								
MOV AX,@DATA								
MOV DS,AX								
MOV SI, OFFSET STRING	04H							
AUSGABE:								
MOV DL, [SI]		61H		42H		43H		44H
CMP DL, 0								
JE GOON								
CMP DL, 61H 'a'								
JL NEXTBUCH								
CMP DL, 7AH 'z'								
JG NEXTBUCH								
SUB DL, 20H				44H				
MOV [SI], DL	41H							
NEXTBUCH:								
INC SI	02H	42H		43H	44H	45H		
JMP AUSGABE								
GOON:								
DEC SI								
. . .								

2/12

Tragen Sie die NACH der Ausführung der einzelnen Befehle in den oben angegebenen Registern vorhandenen Werte in die Tabelle ein.

1. Oktal 635342,413₈

Dual ^{ergänzt} 0011001110101011101010,10001011000₂ ^{ergänzt}

Hexa 3 3 A E 2, 8 7F 8₁₆ 5/6

2. a = 7A3C₁₆

b = 466A₁₆

A = 10

B = 11

C = 12

D = 13

E = 14

F = 15

a) 7A3C

+ 466A

1 1

COA6₁₆ ✓

7A3C

- 466A

1

33D2₁₆ ✓

b) FFFF

- 466A

B995 + 1 = B996

(Über-Komplement)

7A3C

+ B996

1 1 1

33D2₁₆ ✓ 8/8

3. 75,875

75 : 2 = 37, 1

37 : 2 = 18, 1

18 : 2 = 9, 0

9 : 2 = 4, 1

4 : 2 = 2, 0

2 : 2 = 1, 0

1 : 2 = 0, 1



(3. Fortsetzung)

$$\begin{array}{l} 0,875 \cdot 2 = 0,750, 1 \\ 0,750 \cdot 2 = 0,500, 1 \\ 0,5 \cdot 2 = 0, 1 \end{array}$$

Zusammensetzen

1001011,111 ✓

Normieren

~~1001011~~

$$1001011,111 \cong 1,001011111 \cdot 2^6$$

$$127 + 6 = 133_{10}$$

$$133 : 2 = 66, 1$$

$$66 : 2 = 33, 0$$

$$33 : 2 = 16, 1$$

$$16 : 2 = 8, 0$$

$$8 : 2 = 4, 0$$

$$4 : 2 = 2, 0$$

$$2 : 2 = 1, 0$$

$$1 : 2 = 0, 1$$

Exponent 10000101 ✓

Vorzeichen positiv $\rightarrow 0$ ✓

0 10000101 00101111...0
↑
Vorzeichen Exponent Fraktion (mit 0 auf 23 Bit aufgestockt) ✓

11/19

4a)

$$DNF = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}B\bar{C}\bar{D} + A\bar{B}\bar{C}\bar{D} + A\bar{B}C\bar{D} + \bar{A}B\bar{C}D + A\bar{B}C\bar{D} + \bar{A}\bar{B}C\bar{D} + A\bar{B}C\bar{D}$$

b)

AB \ CD	00	01	11	10
00	1	1	X	
01	1	X	X	1
11	1	X	X	1
10		1	X	1

$$= \bar{A}\bar{B}\bar{C} + B +$$

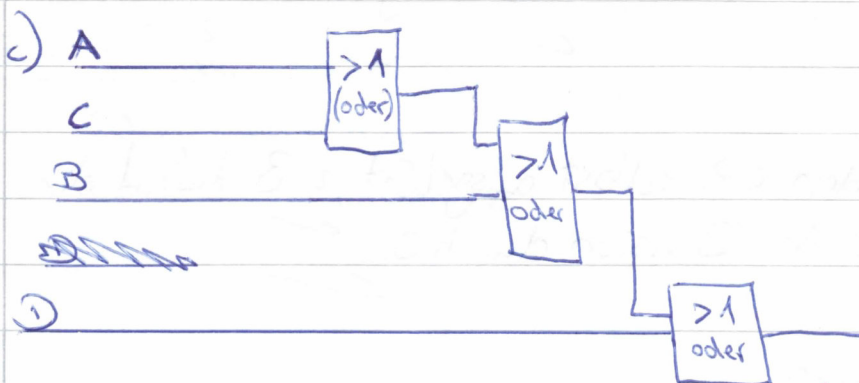
AB \ CD	00	01	11	10
00	1	1	X	
01	1	X	X	1
11	1	X	X	1
10		1	X	1

$$= \bar{A}\bar{C} + B + D + AC$$

$$= \bar{A}\bar{C} + AC + B + D$$

$$= (A \odot C) + B + D$$

f. (-2)



13/15

5a)

- 1) Daten und Befehle werden in einem Speicher gespeichert ✓
- 2) Es gibt nur einen Daten- und Befehlsbus ✓
- 3) Flexibler Speicherverwaltung Aufteilung des Speichers ✓
- 4) geringer Hardwareaufwand ✓

5b) Als „von Neumann'schen Flaschenhals“ bezeichnet man die Situation, dass es durch den einen Daten- / Befehlsbus zur Verlangsamung der Abarbeitung der Eingänge kommt. Der CPU kann die Daten und Befehle nicht so schnell bearbeiten, wie sie ihm erreichen. ✓

c) Durch die Harvard Architektur lässt sich dieses Problem vermeiden, da diese Daten und Befehle getrennt gespeichert und für jedes einen eigenen Bus verfügt. Somit ist ~~ein schneller parallel laufender~~ eine schnelle parallel laufende Abarbeitung möglich. Heutzutage wird oft eine Mischform der beiden Architekturen verwendet. ✓ 8/8 ✓

6. Software-Interrupts ✓

werden vom Programmierer ~~setzt~~ ausgelöst. z.B. INT 21H ✓

Exceptions ✓

werden durch den CPU selbst ausgelöst, z.B. Aufruf der Interrupt Nr. 0 bei der Division durch 0. ✓

Hardware-Interrupts ✓

werden extern, z.B. durch das Drücken einer Taste, über die Hardware-Bauteile ausgelöst ✓ 9/9

7.

a) Rendering ist die Bild darstellung am Rechner aus geometrischen Figuren. ✓

7.b)

1. Application

- Kollisionsberechnung
- Interaktion mit dem Anwender
- Culling
- Zerlegung in Polygone

2. Geometry

- Transformation (Kamera-Perspektiven, Bewegung der Objekte
in den Weltkoordinaten festlegen)
- Lighting (Beleuchtung, Raytracing)
- Projektion (3D-Kamera-Koordination)
- Clipping (abtrennen nicht sichtbarer Teile)
- Screen Mapping (2D-Kamera-Koordination)

3. Rasterizer

- Laden von Texturen
- Abbilden von Polygone

17/17

8) CISC

(Complex Instruction Set Computer)

- 1 Komplexe Instruktionen, Ausführung in mehreren Taktschritten
- 2 Jede Instruktion kann auf den Speicher zugreifen
- 3 Kein oder wenig Pipelining
- 4 Instruktionen werden durch das Mikroprogramm interpretiert
- 5 Instruktionen haben variable Länge
- 6 kompakter Code
- 7 einfacher Speicher Registersatz
- 8 Viele Instruktionen- und Adressspeicherarten
- 9 Komplexität liegt im Mikroprogramm

RISC

(Reduced Instruction Set Computer)

- 1 einfache Instruktionen, Ausführung in einer Taktschritt
- 2 Nur Speicher- und Ladeinstruktionen haben Zugriff auf den Speicher
- 3 intensives Pipelining
- 4 Instruktionen werden durch festverdrahtete Hardware ausgeführt
- 5 Alle Instruktionen mit fester Länge
- 6 weniger kompakter Code
- 7 komplexer Registersatz
- 8 wenige Instruktionen- und Adressspeicherarten
- 9 Komplexität liegt im Compiler

10/10 5/12/12