



Klausur Technische Grundlagen der Informatik
Studiengang Medieninformatik
 H. Linnemann

Freitag, 15. Juli 2011, 08.00 Uhr, Raum B301

- 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	6
Frage 2	8	8
Frage 3	14	14
Frage 4	15	14
Frage 5	8	8
Frage 6	9	9
Frage 7	17	17
Frage 8	12	12
Frage 9	11	11
SUMME	100	99

NOTE: 1,0

- 1.) Wandeln Sie die Oktalzahl 635342,46 in das Dual- und das Hexadezimalsystem!
- 2.) Es sind zwei Hexadezimalzahlen gegeben: $a = 8A3C_{16}$, $b = 746A_{16}$.
 - 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 64,375 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 c!** 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	00	01	11	10	
AB	00	1	1	X	1	
	01	0	X	X	X	1
	11	1	X	X	X	1
	10	1	1	X		1

siehe Lösungsblatt

- 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 ein \$-Zeichen 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, 's' = 24H).

Befehl	Register							
	BX	DL	BX	DL	BX	DL	BX	DL
	1. Durch- lauf	2. Durch- lauf	3. Durch- lauf	4. Durch- lauf				
. . .								
.DATA								
HEINZ DW 0 ;Irgend etwas								
STRING DB 'Acb\$' ;Der OFFSET von String ist 2!								
.CODE								
MOV AX,@DATA								
MOV DS,AX								
MOV BX, OFFSET STRING	02H							
AUSGABE:								
MOV DL, [BX]		41H		63H		62H		24H
CMP DL, 24H		✓		✓		✓		✓
JE WEITER		✓		✓		✓		✓
CMP DL, 61H 'a'		✓		✓		✓		✓
JL NEXTBUCH		✓		✓		✓		✓
CMP DL, 7AH 'z'		✓		✓		✓		✓
JG NEXTBUCH		✓		✓		✓		✓
SUB DL, 20H -20H		✓		43H		42H		✓
MOV [BX], DL		✓	BxL23 =43H	✓	BxL23 =42H	✓		✓
NEXTBUCH:								
INC BX	03H	✓	04H	✓	05H	✓		✓
JMP AUSGABE	✓	✓	✓	✓	✓	✓		✓
WEITER:								
DEC DL		✓						23H
. . .								

11/11

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

8
 24H
 A B C a b c z
 41 42 43 61 62 63 7AH

5.) a) Eigenschaften eines von-Neumann-Rechners:

- 1 Bus für Daten & Befehle ✓
- flexible Speicheranfertigung von Daten & Befehlen ✓
- Anwendung: gängige Home-PCs ✓
- Daten und Befehle in einem Speicher ✓
- sequenzielle Abarbeitung der Befehle ✓

b) Von-Neumann'scher Flaschenhals

Der von-Neumann'sche Flaschenhals entsteht bei einem von-Neumann-Rechner.

Da es nur 1 Bus für Daten und Befehle gibt, muss die CPU auf dem langsamen Speicher "warten", der den Bus blockiert.

c) Flaschenhals vermeiden.

Man kann den von-Neumann-Rechner mit einem Cache oder mit der Harvard-Architektur verbinden, oder ausschließlich eines dieser beiden Architekturen verwenden.

Bei der Harvard-Architektur gibt es beispielsweise einen Bus für Befehle und einen für die Daten. So kann gleichzeitig und schnell auf beide (hier getrennte) Speicher zugegriffen werden.

1)

	6	3	5	3	4	2	4	6
A=10	0	1	1	0	0	1	1	0
B=11	0	1	1	0	1	1	1	0
C=12	0	1	1	0	1	0	0	1
D=13	0	1	1	0	1	0	0	1
E=14	0	1	1	0	1	0	0	1
F=15	0	1	1	0	1	0	0	1

3 3 A E 2 9 8

6/6

2) a)

$$\begin{array}{r} 8A3C_{16} \\ + 746A_{16} \\ \hline FEA6_{16} \end{array} \quad \begin{array}{r} 8A3C_{16} \\ - 746A_{16} \\ \hline 15D2_{16} \end{array}$$

b)

$$\begin{array}{r} FFFF_{16} \\ - 746A_{16} \\ \hline 8B95 \\ + 1 \\ \hline 8B96 \end{array} \quad \begin{array}{r} 8A3C \\ + 8B96 \\ \hline (1)15D2_{16} \end{array}$$

8/8

3) 64,375 ins IEEE754format

1. Vorkommateille $64/2 = 32 \quad R:0$
 $32/2 = 16 \quad R:0$
 $16/2 = 8 \quad R:0$
 $8/2 = 4 \quad R:0$
 $4/2 = 2 \quad R:0$
 $2/2 = 1 \quad R:0$
 $1/2 = 0 \quad R:1$

2. Nachkommateille $0,375 \cdot 2 = 0,75$
 $0,75 \cdot 2 = 1,5$
 $0,5 \cdot 2 = 1$

3. Zusammen setzen 1000 000,011

3. Normalisieren $1,000\ 000\ 011 \cdot 2^6$

4. Offset: $127 + 6 = 133$
 $= 128 + 5$
 $= 1000\ 0000 + 101$
 $= 1000\ 0101$

5. Vorzeichen: + $\rightarrow 0$

6. Zusammensetzen

0 10000101 0000000110...0
 Vorzeichen Exponent Fraction (auf 23 Bit mit 0 aufgefüllt)

4) a) $(\neg D \wedge \neg C \wedge \neg B \wedge A) \vee (\neg D \wedge \neg C \wedge B \wedge A) \vee$
 $(\neg D \wedge \neg C \wedge \neg B \wedge \neg A) \vee (\neg D \wedge C \wedge \neg B \wedge \neg A) \vee$
 $(\neg D \wedge C \wedge \neg B \wedge A) \vee (\neg D \wedge C \wedge B \wedge \neg A) \vee$
 $(\neg D \wedge C \wedge B \wedge A) \vee (D \wedge \neg C \wedge \neg B \wedge \neg A) \vee$
 $(D \wedge \neg C \wedge \neg B \wedge A)$

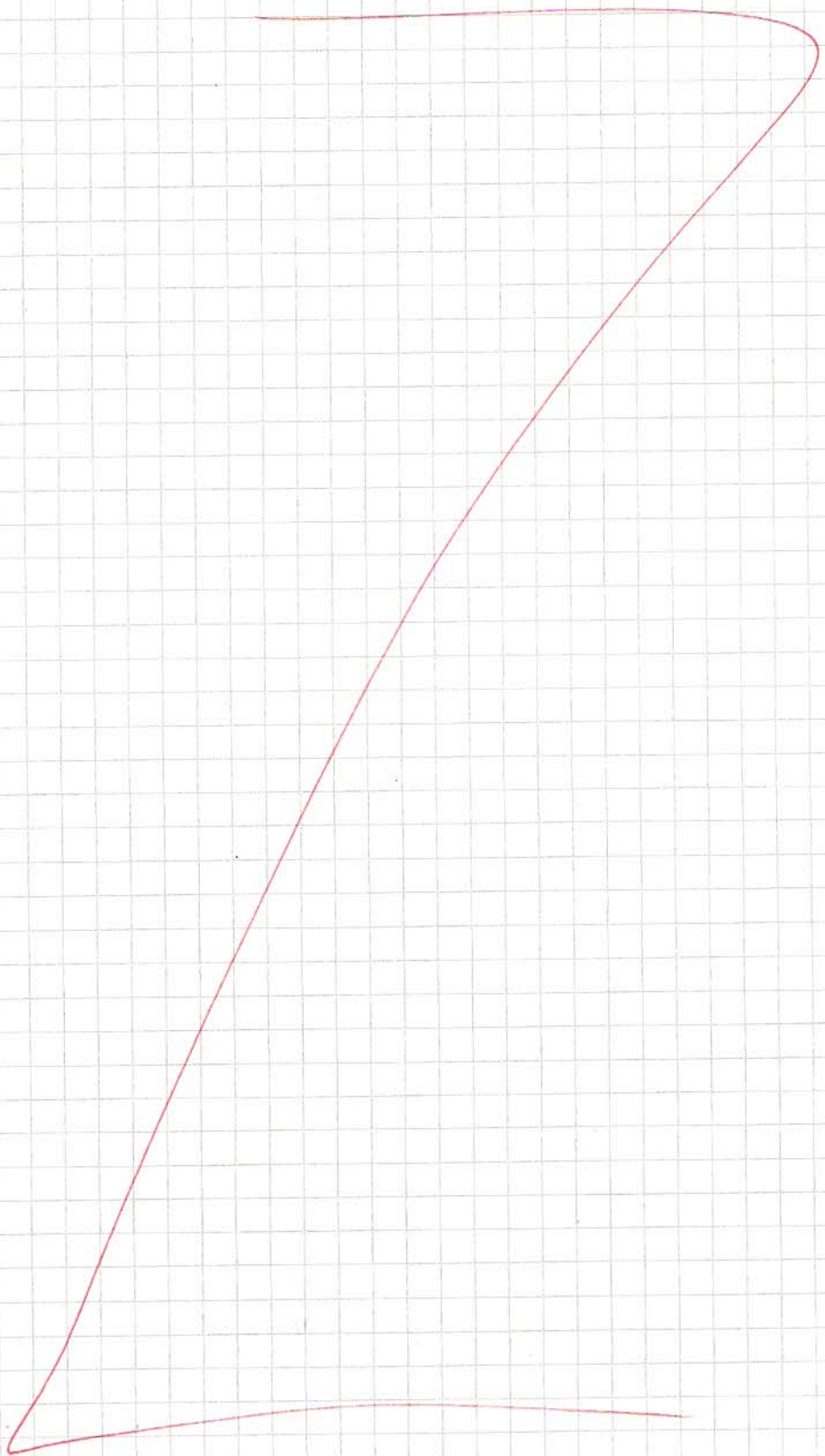
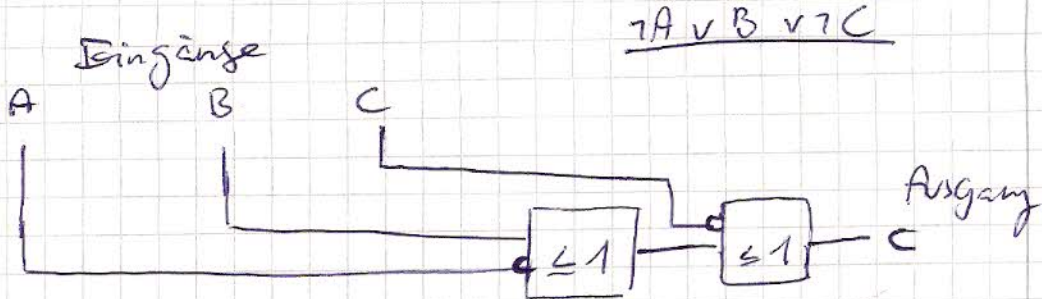
b)

	CD	CD	CD	CD
AB	00	01	11	10
00	1	1	X	1
01	0	X	X	1
11	1	X	X	1
10	1	1	X	1

minimierte Funktion:
 ~~$(\neg A \wedge \neg B) \vee (\neg A \wedge \neg B \wedge \neg C) \vee$~~
 ~~$(\neg C \wedge \neg D \wedge A) \vee (A \wedge B \wedge \neg C)$~~
 ~~$(\neg A \wedge B \wedge \neg C)$~~ (für 0)
 KF für 0 und 1
 $= \neg A \vee \neg C \vee B$ (-1)
 $A \vee B \vee C$

4c) \rightarrow

4c)



7) c) Bilderzeugung am Computer aus geometrischen Figuren ✓

b) Application:

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

Geometry ✓

- Transformation (Kameraperspektive, Bewegung und Position der Objekte in Weltkoordinaten festlegen) ✓
- Lighting (Beleuchtung, Lichteffekte) ✓
- Projection (3D-Koordinaten in 2D-Koordinaten) ✓
- Clipping (Abtrennen nicht sichtbarer Objekte) ✓
- Screen Mapping (2D-Koordinaten) ✓

Rasterizer ✓
17/17

- Einsetzen der Texturen der Polygone ✓
- Sichtbarkeitsberechnung (Z-Buffer) ✓
- Umsetzung der Polygone in Pixel ✓

8) a) Eigenschaften	RISC	CISC
	Reduced Instruction Set Computer	Complex Instruction Set Computer
Anzahl der Register	viele Register ✓	wenige Register ✓
Architektur	LOAD/STORE ✓	Akkumulator & LOAD/STORE ✓
Befehle / Code	weniger komplexe Befehle ✓	komplexe Befehle ✓
Länge der Befehle	immer gleich lang ✓	unterschiedlich lang ✓
Taktung	1 Takt / Befehl ✓	unterschiedlich viele Takte pro Befehl ✓
Befehlssteuerung	meist durch festverdrahtete Hardware ✓	meist durch Mikroprogramme ✓
Pipelining	wenig / kein Pipelining ✓	intensives Pipelining ✓
Zugriff	nur LOAD und STORE können auf Speicher zugreifen ✓	theoretisch können alle Befehle auf den Speicher zugreifen ✓
Deliveriesaufwand	hoch ✓	niedrig ✓

12/12

6) Klassen	Art der Auslösung	Beispiel
Software-Interrupts ✓	durch den Programmierer ("geplant") ✓	INT 21H (Dos-Funktionsaufruf) ✓
Exceptions ✓	durch den Prozessor selbst ✓	INT 0 (bei Division durch 0) ✓
externe Interrupts ✓	durch externe Komponenten ✓	• Maus bewegt / gedrückt ✓ • Taste gedrückt ✓