

Bitte bearbeiten Sie alle Aufgaben im Umfang von insgesamt **80 Punkten** und tragen Sie Ihre Antworten direkt in die Aufgabenblätter ein. Nutzen Sie in dieser Probeklausur auch die Vorlesungsfolien (in der Prüfung geht das nicht).

1. XML und XHTML

(5 P.)

Bei der Überprüfung von XML-Dokumenten mit einer **Document Type Definition** (DTD) unterscheiden wir zwischen **wohlgeformten** und **validen** Dokumenten.

a) Erklären Sie den Unterschied zwischen Wohlgeformtheit und Validität.

1)

2)

1) syntaktisch korrektes XML

2) befolgt zus. die Regeln der DTD

b) XHTML ist eine XML-Sprache, die HTML nachbildet. Nennen Sie zwei Punkte, auf die Programmierer / Web-Entwickler bei XHTML achten müssen, während sie bei HTML keine Rolle spielen.

- Tags in richtiger Reihenfolge schließen

- keine „nur öffnenden“ Tags : $\langle hr \rangle \rightarrow \langle hr / \rangle$

- Groß-/Klein-Schreibung, $\langle LI \rangle \rightarrow \langle li \rangle$

2. Zeichen-Kodierung

(7 P.)

Das Textsatz-Programm LaTeX unterstützt aus historischen Gründen eine eigenwillige Kodierung für deutsche Umlaute: $\backslash "a$ interpretiert das Programm als ä, entsprechend $\backslash "A$ als Ä, $\backslash "o$ als ö usw.; für das scharfe s (ß) gibt es die Kurzform $\backslash "s$. (Der Backslash $\backslash$ ist in LaTeX das allgemeine Zeichen dafür, dass ein Steuerbefehl folgt. Dadurch ist es recht umständlich, einen echten Backslash in den Text zu bekommen: Es gibt dafür den Befehl $\backslash \text{textbackslash}$.)

Bindet man ein spezielles Paket ein, das einige deutsche Anpassungen vornimmt, kann man bei den Umlauten jeweils auf den führenden Backslash verzichten; statt $\backslash "a$ reicht also z. B. $"a$.

a) Zum Starten: Schreiben Sie die folgenden Worte in beiden LaTeX-Notationen auf:

	LaTeX (mit Backslashes)	LaTeX (ohne Backslashes)
müßig	$m \backslash "u \backslash "s i g$	$m "u "s i g$
Eßer :)		
Oeko & Öko	$Oeko \& \backslash "Oko$	$Oeko \& "Oko$
Mämüöm		

b) Nutzt man das Paket für die Kurzvarianten, gibt es ein generelles Problem mit Anführungszeichen – was für ein Problem könnte das sein?

$"a" \rightarrow \ddot{a}$
 $"b" \rightarrow "b"$ } uneinheitlich

c) Die „Herleitung“ dieser LaTeX-Kommandos ist so zu erklären:

\ = Achtung, es folgt ein Befehl

" = Der folgende Buchstabe bekommt ü-Punkte (passt beim „ß“ nicht ganz...)

u = es ist ein u mit ü-Punkten (also ein ü)

Raten Sie, wie die entsprechenden LaTeX-Kommandos aussehen, mit denen sich französische Sonderzeichen (é, è, ê etc.) erzeugen lassen:

é =

è =

ê =

\'e \'e \^e

d) Umlaute werden manchmal als Ä = Ae, ä=ae, Ö=Oe etc. geschrieben, wenn ein Ausgabegerät keine Umlaute beherrscht. Das kann zu unerwarteten Ergebnissen führen. Geben Sie ein Beispiel für ein Wort (und dessen spezielle Schreibweise), bei dem das Ergebnis dieser Umwandlung nicht befriedigend ist.

TUeV

(besser: TUEV)

ue → ue
ü → ue

3. Archivformate (Windows und Linux)

(6 P.)

Windows und Linux verwenden unterschiedliche **Archivformate**, um Dateien zu komprimieren und / oder zu bündeln.

a) Ordnen Sie die folgenden Features den Archivformaten **tar** (Linux), **gzip** (Linux) und **zip** (Windows) zu.

tar gzip zip

☐

☒

☒

Inhalte sind immer komprimiert

☐

☒

☐

Archivdatei enthält genau eine Originaldatei

☒

☐

☒

Archivdatei kann ganze Verzeichnisse enthalten

☒

☐

☐

Inhalte sind nie komprimiert

i.d.R.

? zip

b) Betriebssystem-übergreifende Paketformate müssen Zusatzinformationen (neben Dateiname und Dateieinhalt) speichern. Nennen Sie zwei Arten von Zusatzinformationen (und jeweils das zugehörige Betriebssystem), die beim Entpacken unter einem anderen Betriebssystem Probleme verursachen können.

- Zugriffsrechte Unix ↔ Windows

- Hidden-Bit DOS ↔ Unix

- Datei ist Symlink Unix ↔ Windows

4. Software-Ergonomie

(6 P.)

- a) Kriterien für Benutzerfreundlichkeit und Gebrauchstauglichkeit von Software sind u. a. **Selbstbeschreibungsfähigkeit** und **Lernförderlichkeit**. Nennen und erklären Sie für alle drei Kriterien jeweils ein Beispielverhalten oder -Feature eines Programms, das dieses Kriterium erfüllt.

Selbstbeschreibungsfähigkeit:

Ich weiß immer, wo ich gerade bin, Bsp.: Bearbeitung besteht aus 5 Schritten, sind immer alle genannt, aktueller hervorgehoben

Lernförderlichkeit:

gefährliches Dinge ausprobieren. Bsp.: Es gibt immer ein „Undo“, auch mehrstufig

- b) Nennen Sie (ohne Erklärung) zwei weitere Kriterien aus der Liste der (insgesamt sieben) Kriterien.

- Aufgabenangemessenheit	Erwartungskonform.	Individualisierbarkeit
- Steuerbarkeit	Fehlertoleranz	

5. Von-Neumann-Architektur

(7 P.)

- a) Der Universalrechner (von-Neumann-Rechner) besteht aus fünf Komponenten. Nennen Sie diese fünf Komponenten. (1 P)

Steuer- / Rechen- / Speicher- / Eingabe- / Ausgabe werk

- b) Für ein Programm müssen der Programmcode und die Programmdatei im Speicher vorgehalten werden. Verwenden von-Neumann-Rechner gemeinsame oder getrennte Speichersysteme für Code und Daten? (2 P)

gemeinsam

gemeinsam

- Nennen Sie jeweils einen Vor- und einen Nachteil von ~~getrennten~~ Speichersystemen. (2 P)

+ : Effiziente Speicherutzung (wahlw. Code oder Daten)
- : Sicherheitsproblem, ausführbarer Stack

- c) Was ist die „Pseudo-Harvard-Architektur“? (2 P)

getrennte Level-1-Caches für Code / Daten.
(also auf L1-Cache-Ebene: Harvard)

6. RISC vs. CISC

(6 P.)

Im Folgenden finden Sie sechs fiktive Befehle, deren jeweilige Funktion neben dem Kommando beschrieben ist.

- 1) INCR2 Reg erhöht den Wert im angegebenen Register um 2
- 2) INCR2 Mem erhöht den Wert im Hauptspeicher (an der angeg. Adresse) um 2
- 3) SWAP Mem1,Mem2 vertauscht die Inhalte der Speicherzellen Mem1, Mem2
- 4) CHECK Reg,o1,o2 Prüft Inhalt des angegebenen Registers.
 - Falls Inhalt < 0: relativer Sprung um o1 Byte vorwärts
 - Falls Inhalt > 0: relativer Sprung um o2 Byte vorwärts
 - Falls Inhalt = 0: fortfahren an nächster Adresse
- 5) SAVEALL Mem schreibt den Inhalt sämtlicher Register in den Hauptspeicher (ab Adresse Mem)
- 6) LDBLK Reg,Mem liest einen durch Reg festgelegten 1 KB großen Block von der Festplatte und speichert ihn im Speicher ab Adresse Adr

Kreuzen Sie in der folgenden Tabelle an, welche dieser Befehle es auf einer RISC- bzw. CISC-Architektur geben kann. (Es muss keine konkrete CPU geben, die den jeweiligen Befehl bietet.)

		RISC	CISC
1)	INCR2 Reg	X	X
2)	INCR2 Mem		X
3)	SWAP Mem1,Mem2		X
4)	CHECK Reg,o1,o2	X	X
5)	SAVEALL Mem	X	X
6)	LDBLK Reg,Mem		

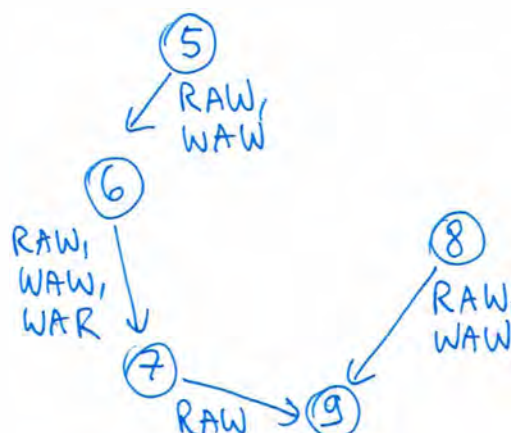
7. Abhängigkeit

(7 P.)

Zeichnen Sie für folgendes Programm den Abhängigkeitsgraph (ohne transitive Pfeile, ohne RAR-Abhängigkeiten). (5 P)

```

1  a    IS    $0
2  b    IS    $1
3  t    IS    $2
4  y    IS    $3
5      MUL   t, a, a
6      SUB   t, t, b
7      SUB   t, t, b
8      MUL   y, b, b
9      SUB   y, y, t
  
```



Geben Sie außerdem an, welche Funktion das Programm berechnet, wenn a und b Argumente, y das Ergebnis und t ein Zwischenwert sind. (2 P)

$$\begin{aligned}
 y = f(a,b) &= b^2 - (a^2 - 2b) \\
 &= b^2 - a^2 + 2b
 \end{aligned}$$



4. Superskalarität

(8/66 Punkte)

a) Was bedeutet „superskalar“ im Vergleich zu „skalar“? (2 P)

skalar: Pro Takt verlässt ein Befehl die Pipeline

superskalar: mehr als einer.

b) Eine CPU-Pipeline bestehe aus 100 Stufen. Fast alle Stufen benötigen zum Erledigen ihrer jeweiligen Aufgabe ca. 1 ns, aber drei der Stufen benötigen ca. 4 ns.

(i) Warum ist das kein vernünftiges Design? (2 P)

Alle Stufen werden auf 4 ns abgebremst, weil die langsamste Stufe den Takt bestimmt.

(ii) Wie lässt sich relativ leicht ein vernünftiges Design aus der bestehenden 100-Stufen-Pipeline entwickeln? (2 P)

So weit möglich, jeweils 4 1-ns-Stufen zu einer 4-ns-Stufe zusammenfassen $\Rightarrow \approx 25$ Stufen mit gleicher Laufzeit

c) In der 5-stufigen RISC-Pipeline kann es zu Kollisionen kommen. Was ist eine Kollision und welche Kollisionen sind hier möglich? (2 P)

Kollision: gleichzeitiger Speicherzugriff (eines der strukturellen Hemmnisse)

In RISC-Pipeline: F (Fetch), M (Memory)

Nur diese eine Variante bei einfacher Pipeline.
Bei superskalarer Pipeline auch M/M

5. Floating-Point-Zahlen

(9/66 Punkte)

Beim Umrechnen von Brüchen in Floating-Point-Zahlen kommt es oft zu Rundungsfehlern. Welche der folgenden Zahlen kann eine CPU exakt als FP-Zahl speichern? Geben Sie zu jeder Zahl eine kurze Begründung an!

	exakt	nicht ex.	Begründung
a) 1,0	X		ganze Zahl
b) 0,1		X	0,1 hat periodische Dualdarst.
c) 1/3		X	1/3 hat keine endl. Dualdarst
d) 0,25	X		$0,25 = 0,01_2$
e) 1/12		X	1/2 und 1/3 haben ähnliche Dualdarstellung (2x Shift)
f) 0,35		X	$= 0,25 + 0,1$



9

6. Stack-Maschine**(5/66 Punkte)**Schreiben Sie ein Programm für die Stack-Maschine, das die folgende Funktion f berechnet:

$$f(a,b,c) = (a^2 + b^2 + c^2) / (a + b + c)$$

Dabei können Sie davon ausgehen, dass a, b, c bereits auf dem Stack liegen (der erste POP-Aufruf holt c vom Stack). Sie können Speicheradressen ab $0x1000$ frei zum Speichern von Zwischenergebnissen verwenden. Das Endergebnis soll nach Ende Ihres Codes auf dem Stack liegen.

POP	1003	(c)	PUSH	1001	(Σ^2, a)
POP	1002	(b)	PUSH	1002	(Σ^2, a, b)
POP	1001	(a)	PUSH	1003	(Σ^2, a, b, c)
PUSH	1001	(a)	ADD		
PUSH	1001	(a, a)	ADD		(Σ^2, Σ)
MULT		(a^2)	DIV		(Σ^2 / Σ) ✓
PUSH	1002	(a^2, b)			
PUSH	1002	(a^2, b, b)			
MULT		(a^2, b^2)			
PUSH	1003	(a^2, b^2, c)			
PUSH	1003	(a^2, b^2, c, c)			
MULT		(a^2, b^2, c^2)			
ADD		($a^2, b^2 + c^2$)			
ADD		($a^2 + b^2 + c^2$)			

7. Virtuelle Adressen / Paging**(10/66 Punkte)**

Eine CPU arbeitet mit folgenden Werten:

- Seitengröße: 4 KByte $\rightarrow$ 12 Bit Offset, $(32-12) = 20$ Bit Seiten-Nr.
- 32 Bit lange virtuelle Adressen
- Seitentableneinträge sind 4 Byte lang

a) Wie ist eine virtuelle Adresse aufgebaut (welche Bits der Adresse haben welche Bedeutung)? (2)



Zeichnen Sie die Unterteilung hier ein und beschriften Sie die Abschnitte geeignet.

b) Wie groß ist die gesamte Seitentabelle? (5 P)

Ein Eintrag: 4 B

Anzahl Einträge: $2^{20} = 1M \Rightarrow 4MByte$

9. Stack-Maschine

(5 P.)

Schreiben Sie ein Programm für die Stack-Maschine, das die folgende Funktion f berechnet:

$$f(a,b,c) = (a^2+b^2+c^2) / (a+b+c)$$

Dabei können Sie davon ausgehen, dass a , b , c bereits auf dem Stack liegen (der erste POP-Aufruf holt c vom Stack). Sie können Speicheradressen ab $0x1000$ frei zum Speichern von Zwischenergebnissen verwenden. Das Endergebnis soll nach Ende Ihres Codes auf dem Stack liegen.

siehe anderes Blatt

10. Reorder Buffer

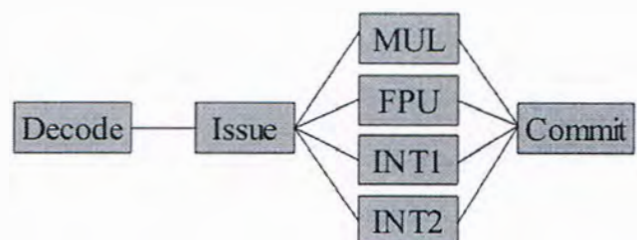
(9 P.)

Betrachten Sie die folgenden vier Befehle:

- 1 FADD \$5,\$1,\$2
- 2 FSUB \$6,\$1,\$2
- 3 FMUL \$1,\$5,\$6
- 4 FADD \$1,\$1,1

Die Befehle FMUL, FADD und FSUB benötigen jeweils vier Zyklen auf der MUL- bzw. FPU-Ausführungseinheit.

Zeigen Sie, wie diese Befehle auf einem System mit Reorder Buffer (zwei Zuteilungen pro Takt) ausgeführt werden; zeichnen Sie dazu die Inhalte des Reorder Buffers zu den 17 Zeitpunkten t_0 bis t_{16} ein.



0

			Operanden				
Befehl	unit	Status	Q _Y	Q _Z	V _Y	V _Z	Ergebnis
FADD	FPU	D1	\$1	\$2			\$5 (u.)

1

			Operanden				
Befehl	unit	Status	Q _Y	Q _Z	V _Y	V _Z	Ergebnis
FADD	FPU	D2					\$5 (u.)

2

			Operanden				
Befehl	unit	Status	Q _Y	Q _Z	V _Y	V _Z	Ergebnis
FADD	FPU	D3					\$5 (u.)

3

			Operanden				
Befehl	unit	Status	Q _Y	Q _Z	V _Y	V _Z	Ergebnis
FADD	FPU	D4					\$5 (u.)

4

			Operanden				
Befehl	unit	Status	Q _Y	Q _Z	V _Y	V _Z	Ergebnis
FADD	—	✓					\$5
FSUB	FPU	D1	\$1	\$2			\$6 (u.)
FMUL	MUL	II	\$5*	—		FSUB	\$1 (u.)

5

			Operanden				
Befehl	unit	Status	Q _Y	Q _Z	V _Y	V _Z	Ergebnis
FSUB	FPU	D2					\$6 (u.)
FMUL	MUL	II	\$5	—		FSUB	\$1 (u.)

! \$5
nicht
mehr im
Reorder Buf.

6

			Operanden				
Befehl	unit	Status	Q _Y	Q _Z	V _Y	V _Z	Ergebnis
FSUB	FPU	D3					\$6 (u.)
FMUL	MUL	II	\$5*	—		FSUB	\$1 (u.)

7

			Operanden				Ergebnis
Befehl	unit	Status	Q _Y	Q _Z	V _Y	V _Z	
FSUB	FPU	4					\$6 (u.)
FMUL	MUL	11	\$5	—		FSUB	\$1 (u.)

8

			Operanden				Ergebnis
Befehl	unit	Status	Q _Y	Q _Z	V _Y	V _Z	
FSUB	—	✓					\$6*
FMUL	MUL	1	\$5	\$6*			\$1 (u.)
FADD	FPU	11	—	×	FMUL		\$1 (u.)

9

			Operanden				Ergebnis
Befehl	unit	Status	Q _Y	Q _Z	V _Y	V _Z	
FMUL	MUL	2					\$1 (u.)
FADD	FPU	11	—		FMUL		\$1 (u.)

10

			Operanden				Ergebnis
Befehl	unit	Status	Q _Y	Q _Z	V _Y	V _Z	
FMUL	MUL	3					\$1 (u.)
FADD	FPU	11	—		FMUL		\$1 (u.)

11

			Operanden				Ergebnis
Befehl	unit	Status	Q _Y	Q _Z	V _Y	V _Z	
FMUL	MUL	4					\$1 (u.)
FADD	FPU	11	—		FMUL		\$1 (u.)

12

			Operanden				Ergebnis
Befehl	unit	Status	Q _Y	Q _Z	V _Y	V _Z	
FMUL	—	✓					\$1*
FADD	FPU	1	\$1*				\$1 (u.)

13

			Operanden				Ergebnis
Befehl	unit	Status	Q _Y	Q _Z	V _Y	V _Z	
FADD	FPU	2					\$1 (u.)

14

			Operanden				Ergebnis
Befehl	unit	Status	Q _Y	Q _Z	V _Y	V _Z	
FADD	FPU	D3					\$1 (v)

15

			Operanden				Ergebnis
Befehl	unit	Status	Q _Y	Q _Z	V _Y	V _Z	
FADD	FPU	D4					\$1 (u)

16

			Operanden				Ergebnis
Befehl	unit	Status	Q _Y	Q _Z	V _Y	V _Z	
FADD	—	✓					\$1

11. PC als Arbeitsplatz

(4 P.)

Welche der folgenden Aussagen sind korrekt? Bitte die richtigen Aussagen ankreuzen.

- ☐ Computer-Algebra-Systeme (CAS) haben besondere Stärken in der numerischen Mathematik: Sie berechnen Lösungen durch Näherungsverfahren.
- ☐ In der Versionsverwaltung entsteht ein „Branch“, wenn die aktuellste Version einer Datei bearbeitet und in veränderter Form eingchecked wird.
- ☒ „Branches“ kann man durch eine „Merge“-Operation wieder zusammenführen.
- ☒ Beim Erstellen von LaTeX-Dokumenten konzentrieren Autoren sich in erster Linie auf den Inhalt und nicht auf die Form.

12. Verteilte Systeme

(10 P.)

- a) Erklären Sie den Unterschied zwischen *RPC* und Asynchronous RPC und geben Sie ein Beispiel für den sinnvollen Einsatz von Asynchr. RPC.
- b) Amdahls Gesetz: Ein Programm bestehe zu $\alpha = 50\%$ aus nicht parallelisierbarem Code, und durch Einsatz mehrerer CPUs sei bereits ein Speed-up von 1,95 erreicht. Ist es in dieser Situation sinnvoll, weitere CPUs hinzuzufügen? Begründen Sie Ihre Antwort. $\frac{1}{\alpha} = \frac{1}{0,5} = 2 = \text{max. Speed-up}$
- c) Erläutern Sie den Begriff der *Bit-Level-Parallelität*.
- d) Nennen Sie zwei zentrale Unterschiede zwischen einem Grid und einer Cloud.

- heterogen vs. einheitliche HW
- weite räumliche Verteilung vs. RZ
- i.d.R.: phys. Maschinen vs. virtuelle Maschinen

→ #Bits, die in einer Instruction verarb. werden; 16-/32-/64-bit-CPU's