## Prüfungsklausur Computersysteme (25211)

Teil II: Kurs 1609

SS 2017

## Prof. Dr. W. Schiffmann 23.09.2017

Studierende	r:	
Name:	Matrikelnu	mmer:
Aufgabenüb	ersicht:	
Aufgabe II-1:	Fragen zur Rechnerarchitektur	(10 Punkte)
Aufgabe II-2:	Gleitkommadarstellung	(10 Punkte)
Aufgabe II-3:	Code-Analyse	(15 Punkte)
Aufashe II-A	Speicherverwaltung	(15 Punkte)

### Punkteverteilung:

	1	2	3	4	Summe
bearbeitet					
max. Punktzahl	10	10	15	15	50
erreichte Punktzahl					

#### **Hinweis:**

Die vier Klausuraufgaben befinden sich auf insgesamt 9 Seiten: II-1 bis II-9. Überprüfen Sie bitte die Vollständigkeit Ihres Klausurexemplars!

# Aufgabe II-1: Fragen zur Rechnerarchitektur (10 P)

a) Geben Sie an, ob die folgenden Aussagen wahr (W) oder falsch (F) sind:
W F
1) Befehlspipelining ist eine Mikroarchitekturtechnik.
○ ○ 2) Statisches Befehlsscheduling ist eine Mikroarchitekturtechnik.
O 3) Die Memory Management Unit (MMU) wandelt physikalische in logische Adressen um.
O 4) Bei der FIFO-Verdrängungsstrategie wird der am längsten nicht verwendete Cache-Eintrag zuerst gelöscht.
○ ○ 5) Ein RISC-Prozessor hat deutlich mehr Register als ein CISC-Prozessor.
○ ○ 6) Ein Permutationsnetz ist ein statisches Verbindungsnetz.
O 7) Bei der Von-Neumann-Architektur sind Programmcode und Daten in ver- schiedenen Speichern abgelegt.
O 8) Im Little-endian-Format zeigt die Speicheradresse auf das niedrigstwertige Byte eines Speicherwortes.
b) Geben Sie an, was das Lokalitätsprinzip besagt:
c) Nennen Sie die verschiedenen Typen von Datenabhängigkeiten und beschreiber Sie kurz, wann diese vorliegen.
d) Caben Sie die Bedeutung der Begriffe Cache-Kohärenz und Cache-Konsistenz an

# Aufgabe II-2: Gleitkommadarstellung (10 P)

Gegeben sei die Dezimalzahl  $Z_{10} = -11,8125.$ 

a) Stellen Sie die Zahl  $Z_{10}$  als gebrochene, normalisierte Zahl  $Z_{32}$  im 32-bit-Format des IEEE-754-Standards dar und tragen Sie dazu die entsprechenden Werte für Vorzeichen, verschobenen Exponenten und Mantisse in das folgende Schema ein:

$$Z_{32} = (-1)^{\dots \dots \cdot 2^{(\dots \dots \cdot)_{10}} \cdot (\dots \dots \cdot \dots \cdot \dots \cdot \dots \cdot \dots \cdot )_2$$

b) Tragen Sie die Zahl  $Z_{32}$  in binärer Darstellung in den folgenden Bitrahmen ein und kennzeichnen sowie bezeichnen Sie die unterscheidbaren Bitfelder.

313029282	2726	2524	123	222	120	) 19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													9				N								
				an i	37																				
		Tra	gen	Sie	hie	r d	ie l	Bez	zeio	chn	un	ger	ı d	er	Bit	fel	de	r ei	n						

c) Geben Sie die Zahl  $\mathbb{Z}_{32}$  als Hexadezimalzahl  $\mathbb{Z}_{16}$  an.

$$Z_{16} = \dots$$

d) Gegeben seien nun die Zahlen  $Z1_{10} = 9,75$  und  $Z2_{10} = 13,125$ . Wandeln Sie diese Zahlen in das IEEE-754-Format um und bilden Sie die Summe dieser beiden Zahlen. Wie erfolgt die Angleichung der Charakteristik? Die Nebenrechnung in binärer Form (Rechnung im IEEE-754-System) ist erforderlich. Führen Sie die Addition aus und tragen Sie das Endergebnis ein.

3130292827263	25242	23222	120	191	817	16	15	141	312	11	10 9	8	7	6	5	4	3	2	1	0
																		Ŋ.		

Hinweis:

Die Indizes  $\cdots_2$ ,  $\cdots_{10}$ ,  $\cdots_{16}$  und  $\cdots_{32}$  kennzeichnen jeweils Zahlen im Binär-, Dezimalsowie Hexadezimal-System bzw. im 32-Bit-IEEE-Format.

# Aufgabe II-3: Code-Analyse (15 P)

Gegeben sei der unten stehende DLX-Assembler-Code.

Zeile	Marke	Anweis	ung				
1 2 3 4 5 6 7		LA LW LW LW AND LB BEQZ	R1, R3, R4, R5, R8, R2,	in mask op RO, (R1)	2 R0		
8 9 10 11	m1:	AND BEQZ XOR ADDI	R7, R7, R2, R2,	R2, m5 R2, R2,			
12 13 14	m5:	JAL LB BNEZ	m3 R2, R2,	(R1) m1	+		
15 16	m2: end:	SW J	out, end	R8			
17 18 19 20	m3:	SLE BEQZ SUB J	R6, R6, R2, m3	R5, m4 R2,			
21 22 23	m4:	SLLI OR JR	R8, R8, R31	R8, R8,		3	
24 25 26 27 28	<pre>in: op: mask1: mask2: out:</pre>	.byte .word .word .word	0x0A 0x80 0xFF	0x0E 00000 FFFFF ADBEE	F	0x02	0x00

### Hinweise:

- Immediate-Werte werden mit einem Doppelkreuz (#) und Hexadezimalzahlen mit einem einleitenden 0x gekennzeichnet.
- Bei Store-Befehlen gibt der erste Operand das Ziel im Speicher, der zweite Operand die Quelle (ein Register) an. Bei Load-Befehlen gibt der erste Operand das Ziel (ein Register), der zweite Operand die Quelle im Speicher an.
- Der Befehl LA lädt die Speicheradresse der angegebenen Marke in ein Register.
- Die vier Bytes an der Speicherstelle in: sind als Ganzzahlen im Zweierkomplement zu betrachten.

a)	Beschreiben Sie in einem Satz folgende Assembler-Anweisungen (vgl. Beispiel am Ende dieses Aufgabenteils!):
	Zeile 8: AND R7, R2, R3:
	Zeile 9: BEQZ R7, m5:
	Zeile 21: SLLI R8, R8, #0x8:

### Beispiel:

Zeile 2 (LW R3, mask1) Load Word: Diese Anweisung lädt das Wort an der Speicherstelle mask1 in das Register R3.

FernUniversität in Hagen

Prüfungsklausur Computersysteme (1609)

Seite II-5

Seite für Notizen:

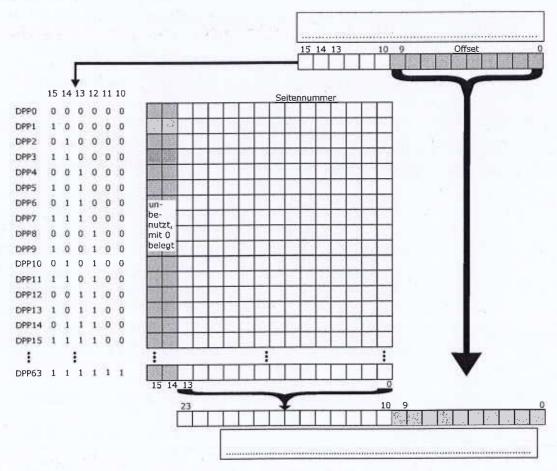
b)	Erläutern Sie, was der Befehl in Zeile 12 (JAL m3) bewirkt und wie oft diese im Programmverlauf aufgerufen wird. Wie hängt der Befehl (JR R31) in Zeile 23 mit diesem zusammen?
	***************************************
	α
	***************************************
c)	Nennen Sie die in der Zeile 6 (LB R2, (R1)+) verwendete Adressierungsart und erläutern Sie diese.
	***************************************
	***************************************
	***************************************
e)	Beschreiben Sie, was das obige Programm funktional macht und welcher Wert bei Programmende an der Speicherstelle out abgespeichert wird.
	**************************************
	vivos

# Aufgabe II-4: Speicherverwaltung (15 P)

Ein einfacher Mikroprozessor mit einem 16-Bit-Datenbus und einem 24-Bit-Adressbus unterstütze eine rudimentäre Form der "virtuellen Speicherverwaltung" mit Seiteneinteilung, wie sie in der unten stehenden Zeichnung skizziert ist. Dazu stehen dem Prozessor 64 16-Bit-Register DPPi, i=0,...,63, (Datenseiten-Zeiger – Data Page Pointer) zur Verfügung, in denen jedoch nur die unteren 14 Bits zur Auswahl von bis zu vier aktuell benötigten Seiten ausgewertet werden. Die in einem Ladebefehl:

LD R, DPPi:<Offset> ("Lade Register R mit dem Inhalt der adressierten Speicherzelle")

angegebene "virtuelle" Adresse ist 16 Bits lang. Ihre höchstwertigen sechs Bits A15, ..., A10 selektieren das verwendete DPPi-Register. In dieser Aufgabe werden ausschließlich die ersten 16 DPP-Register (DPP0, ..., DPP15) betrachtet. Die Speicheradresse (physikalische Speicheradresse) ergibt sich aus der "Verkettung" (Konkatenation) der 14-Bit-Seitennummer im gewählten DPPi-Register und den unteren 10 Bits der virtuellen Adresse als Offset in der gewählten Seite.



	a)	Tragen Sie die Bezeich	in die bei nung der	den mit P jeweiliger	unkten [] g n Adresse ein	ekennzeichi	ieten Felder	der Abbildung
	b)	Wie groß s	ind die se	lektierbar	en Seiten?			
		Größe (in 1	kByte):		*****			
								,
							1.7	
	۵)							
	c)	sprochen w	erscniede: erden?	ne Seiten	konnen im pl	ıysikalische	a Speicher n	naximal ange-
		Anzahl:						
	d) 1	Wie groß is	t der gesa	mte ansp	rechbare phys	ikalische Sı	oeicher maxi	imal?
		Größe (in N				- 220 mil 4		5.10.00
	1	Begründu	ng:					
	d d	lie Register	DPP0, .	Adresse, DPP18	(in hexadezin 5 wie in der n	aler Form	wird ausge en Tabelle b	egeben, wenn elegt sind:
П	)PP(	)   \$05FA	DPP8	\$001A				
_	PP1	10022	DPP9	\$3CDB	-			
I	PP2		DPP10	\$12B4				
Γ	PP3	\$1AA2	DPP11	\$077A				
	PP4		DPP12	\$0F77	1			
_	PP5		DPP13	\$0666	1			
	PP6		DPP14	\$3314				
D	PP7	\$3A39	DPP15	\$0FEE				
					O angesproche			
							*********	
	H	erleitung						

FernUniversität in Hagen

Prüfungsklausur Computersysteme (1609)

Seite II-9

Seite für Notizen: