

Prüfungsklausur Computersysteme (25211)
Teil 1609
SS 2016

Prof. Dr. W. Schiffmann,

17.09.2016

Inhaltsverzeichnis

Aufgabe II-1: Fragen zur Rechnerarchitektur (5 P)	2
Aufgabe II-2: Gleitkommadarstellung (8 P)	3
Aufgabe II-3: Code-Analyse (17 P)	6
Aufgabe II-4: Speicherverwaltung (17 P)	10
Aufgabe II-5: Parallelverarbeitung (3 P)	13

Aufgabe II-1: Fragen zur Rechnerarchitektur (5 P)

Anhand der nachfolgenden Fragen sollen die Unterschiede zwischen Architektur- und Mikroarchitekturstechniken erarbeitet werden.

a) Geben Sie an, ob die folgenden Aussagen wahr (W) oder falsch (F) sind:

W F

- 1) Befehlspipelining ist eine Architekturtechnik.
- 2) Superskalare Prozessoren besitzen eine spezielle Mikroarchitektur.
- 3) Architekturstechniken müssen durch entsprechende Systemsoftware (z.B. Compiler) unterstützt werden.
- 4) Mikroarchitekturen können ausschließlich durch die verfügbaren Maschinenbefehle, Adressierungsarten und für den Programmierer sichtbaren Register beschrieben werden.
- 5) Renaming ist eine Mikroarchitekturstechnik.
- 6) Dynamisches Befehlsscheduling in Superskalarprozessoren ist eine Architekturstechnik.

b) Ordnen Sie den folgenden Prozessortypen zu, ob sie hauptsächlich auf einer Architekturstechnik (A) oder einer Mikroarchitekturstechnik (M) basieren:

Prozessortyp	A	M
Mikroprogrammierter CISC-Prozessor	<input type="radio"/>	<input type="radio"/>
Skalarer RISC-Prozessor	<input type="radio"/>	<input type="radio"/>
Superskalarer RISC-Prozessor	<input type="radio"/>	<input type="radio"/>
VLIW-Prozessor	<input type="radio"/>	<input type="radio"/>

Aufgabe II-2: Gleitkommadarstellung (8 P)

Gegeben sei die Dezimalzahl $Z_{10} = 65,6875$.

- 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:

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

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Tragen Sie hier die Bezeichnungen der Bitfelder ein																															

- c) Geben Sie die Zahl Z_{32} als Hexadezimalzahl Z_{16} an.

$$Z_{16} = \dots\dots\dots$$

- d) Gegeben seien nun die Zahlen $Z1_{10} = 12,75$ und $Z2_{10} = 18,25$. 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.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Hinweis:

Die Indizes \dots_2 , \dots_{10} , \dots_{16} und \dots_{32} kennzeichnen jeweils Zahlen im Binär-, Dezimal- sowie Hexadezimal-System bzw. im 32-Bit-IEEE-Format.

Aufgabe II-3: Code-Analyse (17 P)

Gegeben sei der unten stehende DLX-Assembler-Code. Die Werte im Speicher an den Stellen *op1* und *op2* seien vorzeichenlose Integer-Zahlen.

Zeile	Marke	Anweisung
1		LW R3, mask
2		LW R4, op1
3		LW R5, op2
4		AND R8, R0, R0
5	l1:	BEQZ R4, m1
6		JAL func
7		SLL R6, R5, R2
8		ADD R8, R8, R6
9		ADDI R7, R0, #0x1
10		SLL R7, R7, R2
11		XOR R7, R7, R3
12		AND R4, R4, R7
13		J l1
14	m1:	SW dest R8
15	m2:	J m2
16	func:	SW (R29)- R4
17		AND R2, R0, R0
18	l2:	ADDI R2, R2, #0x1
19		SRLI R4, R4, #0x1
20		BNEZ R4, l2
21		SUBI R2, R2, #0x1
22		LW R4, +(R29)
23		JR R31
24	op1:	.word 0x15
25	op2:	.word 0xa
26	dest:	.word 0xdeadbeef
27	mask:	.word 0xffffffff

Hinweise:

- Der Stackzeiger wird in Register R29 verwaltet.
- Immediate-Werte werden mit einem Doppelkreuz (#) gekennzeichnet.
- Hexadezimalzahlen werden mit einem einleitenden 0x gekennzeichnet.
- Das Register R4 wird für Übergabeparameter, das Register R2 für Rückgabewerte von Funktionen verwendet.
- 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.

a) Erklären Sie in einem Satz folgende Assembler-Anweisungen (vgl. Beispiel am Ende dieses Aufgabenteils!):

Zeile 7: SLL R6, R5, R2:

.....
.....

Zeile 12: AND R4, R4, R7:

.....
.....

Zeile 20: BNEZ R4, 12:

.....
.....

Beispiel:

Zeile 1: LW R3, mask:

Load Word: Diese Anweisung lädt das 32-Bit-Wort an der Speicherstelle mask in das Register R3.

- b) Welche Boole'sche Operation wird durch folgende Anweisung nachgebildet, die nicht explizit Teil des DLX-Sprachumfangs ist? (Hinweis: Beachten Sie den Inhalt des Registers R3!)

Zeile 11: XOR R7, R7, R3

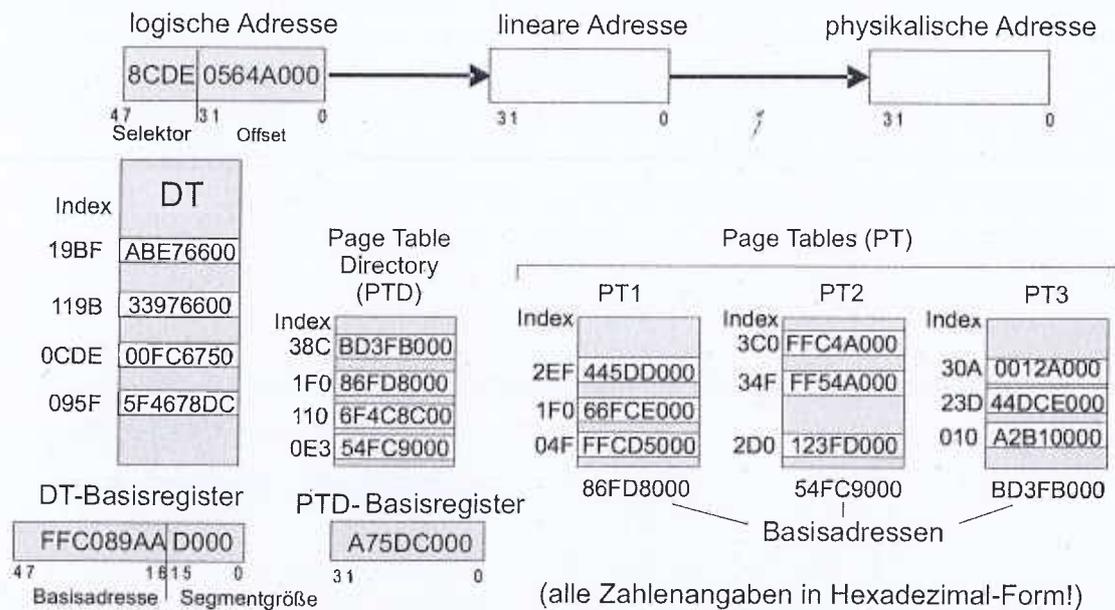
- c) Auf welche Speicherstelle verweist der Stack-Zeiger allgemein während der Ausführung des Programms? Was passiert vor bzw. nach einem schreibenden Stack-Zugriff?

- d) Erläutern Sie, was die Funktion *func* berechnet.

- e) Beschreiben Sie in **einem** Satz, was das obige Programm macht. (Hinweis: Hier wird eine Funktion nachgebildet, für die im DLX-Assembler eine eigene Instruktion existiert.)

Aufgabe II-4: Speicherverwaltung (17 P)

Betrachten Sie die Speicherabbildungsfunktion der x86-Prozessoren im untenstehenden Bild. Es zeigt für einen aktiven Prozess einen Ausschnitt der Speicherbelegung mit der Deskriptor-Tabelle DT (*Descriptor Table*), dem Seitentabellen-Verzeichnis PTD (*Page Table Directory*) und drei Seitentabellen PT_i (*Page Table*). Außerdem sind das DT-Basisregister und das PTD-Basisregister angegeben.



- a) Geben Sie die einzelnen Schritte zur Berechnung der linearen Adresse aus der im Bild vorgegebenen logischen Adresse an! Der Index in die DT besteht aus den höchstwertigen 13 Bits des Selektors. Die Einträge in der DT sind jeweils 8 Byte lang. Tragen Sie die lineare Adresse ins oben stehende Bild ein!

.....

.....

.....

.....

.....

.....

- b) Geben Sie die einzelnen Schritte zur Berechnung der physikalischen Adresse PA aus der unter a) berechneten linearen Adresse an! Gehen Sie dabei von einer Seiten/Tabellengröße von 4 kByte aus. Der Index in die PTD besteht aus den höchstwertigen 10 Bits der linearen Adresse, der Index in die PTi aus den mittleren 10 Bits. Tragen Sie die physikalische Adresse ins oben stehende Bild ein!

.....

.....

.....

.....

.....

- c) Bestimmen Sie die Anfangsadressen AA für die folgenden Tabelleneinträge (Rechnung angeben!):

- Eintrag Nr. 10 in der DT:

$$AA = \dots\dots\dots = \$ \dots\dots\dots$$

- Eintrag Nr. 64 in der PTD:

$$AA = \dots\dots\dots = \$ \dots\dots\dots$$

- Eintrag Nr. 16 in der PT3:

$$AA = \dots\dots\dots = \$ \dots\dots\dots$$

- d) Welche Adresse BA besitzt das Byte Nr. 80 in der durch den Eintrag Nr. 16 in PT3 selektierten Speicherseite (Rechnung angeben!).

$$BA = \dots\dots\dots = \$ \dots\dots\dots$$

Aufgabe II-5: Parallelverarbeitung (3 P)

Ein Algorithmus A habe eine sequentielle Laufzeit von a Zeiteinheiten (ZE). Eine parallele Version dieses Algorithmus habe folgende Eigenschaften: Die Laufzeit a kann beliebig fein parallelisiert werden, indem jeder Prozessor einen Teil der Daten bearbeitet. Wegen der zusätzlichen Synchronisation der Prozessoren verlängert sich der parallele Anteil um 5%. Darüber hinaus kommt pro Prozessor ein sequentieller Anteil von 0,015 ZE hinzu. Für den Broadcast-Nachrichtenversand ergibt sich eine konstante Verlängerung der Laufzeit um 15 ZE.

- a) Stellen Sie die Gleichung für die Berechnung der Laufzeit auf einem Parallelsystem in Abhängigkeit von der Zahl der verfügbaren Prozessoren auf.
- b) Geben Sie die Speedup-Funktion nach Amdahl in Abhängigkeit von der Anzahl der verfügbaren Prozessoren an. Vereinfachen Sie diese Gleichung.

Anmerkung: Leiten Sie Ihre Ergebnisse her.