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)	Superskalarität ist eine Mikroarchitekturtechnik.
◯	◯	2)	VLIW ist eine Mikroarchitekturtechnik.
◯	◯	3)	Der *Translation Lookaside Buffer* (TLB) ist ein vollassoziativer Cache, der die Segmentierung beschleunigt.
◯	◯	4)	Beim Durchschreibeverfahren wird sowohl der Cache-Eintrag als auch der Speicher verändert.
◯	◯	5)	Ein System ist cache-kohärent, wenn bei einem Lesezugriff immer der Wert des ersten Schreibzugriffs geliefert wird.
◯	◯	6)	Durch dynamische Sprungvorhersagetechniken werden Fehlspekulationen vermieden.
◯	◯	7)	Bei der Harvard-Architektur sind Programmcode und Daten in verschiedenen Speichern abgelegt.
◯	◯	8)	Ein superskalarer Prozessor zeichnet sich durch seine hohe Anzahl an Pipeline-Stufen und der daraus resultierenden hohen Taktrate aus.
◯	◯	9)	Bei der befehlszählerrelativen Adressierung wird der Befehlszähler (PC) bei einem 32-Bit-Prozessor mit byteadressierbarem Speicher normalerweise nach jedem Befehl um `+4` erhöht.
◯	◯	10)	Ein virtueller Cache befindet sich funktional zwischen Prozessor und Speicherverwaltung (*Memory Management Unit*, MMU).

b) Nennen Sie die im Kurs vorgestellten **fünf** Pipeline-Stufen und beschreiben Sie kurz, was in den einzelnen Stufen passiert.

c) Geben Sie zwei **typische** Eigenschaften eines RISC-Prozessors an.

d) Benennen Sie die im Kurs vorgestellten **drei** Cache-Organisationsformen und geben Sie an, in welche Teile die Speicheradresse jeweils zerlegt wird (Tag, Index, Byteauswahl).

Lösungsvorschläge

a) Die Aussagen 1, 4, 7, 9 und 10 sind wahr.

b) IF, ID, EX, MEM und WB (siehe Kurstext).

c) Typische Eigenschaften eines RISC-Prozessors:

- viele universell nutzbare Register
- einheitliche Befehlslänge
- fest verdrahtete Befehle (keine Mikroprogrammierung)
- wenige Befehls- und Adressierungsarten
- Load/Store-Architektur bzw. Register-Register-Architektur
- Befehlspipelining

d) Cache-Organisationsformen:

- vollassoziativer Cache (Tag, Byteauswahl)
- direktabgebildeter Cache (Tag, Index, Byteauswahl)
- n-fach satzassoziativer Cache (Tag, Index, Byteauswahl)

Aufgabe II-2: Gleitkommadarstellung (10 P)

Gegeben sei die Dezimalzahl $Z_{10} = 9{,}5625$.

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:

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

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.

$Z_{16} =$ .....

d) Gegeben seien nun die Zahlen $Z1_{10} = 0{,}625$ und $Z2_{10} = 4{,}5$. 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 $\cdots_2$, $\cdots_{10}$, $\cdots_{16}$ und $\cdots_{32}$ kennzeichnen jeweils Zahlen im Binär-, Dezimal- sowie Hexadezimal-System bzw. im 32-Bit-IEEE-Format.

Lösungsvorschläge

a) $Z_{32} = (-1)^0 \cdot 2^{(130)_{10}} \cdot (1 \bullet 00110010..0)_2$

b) Die Bezeichnungen sind s: Vorzeichen, e: versch. Exponent, m: Mantissenteil.

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
0	1	0	0	0	0	0	1	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
s	e								m																						

c) $Z_{16} = 41190000_{16}$

d) Der erste Schritt besteht darin, die beiden gegebenen Zahlen in das IEEE-754-Format umzuwandeln. Folgende binäre Darstellung ergibt sich dabei:

$Z1_{32} = 0\ 01111110\ 0100...0$
$Z2_{32} = 0\ 10000001\ 0010...0$

Der Exponent von $Z1$ wird zum Angleichen um drei erhöht. Die Mantisse wird entsprechend um drei nach rechts verschoben.

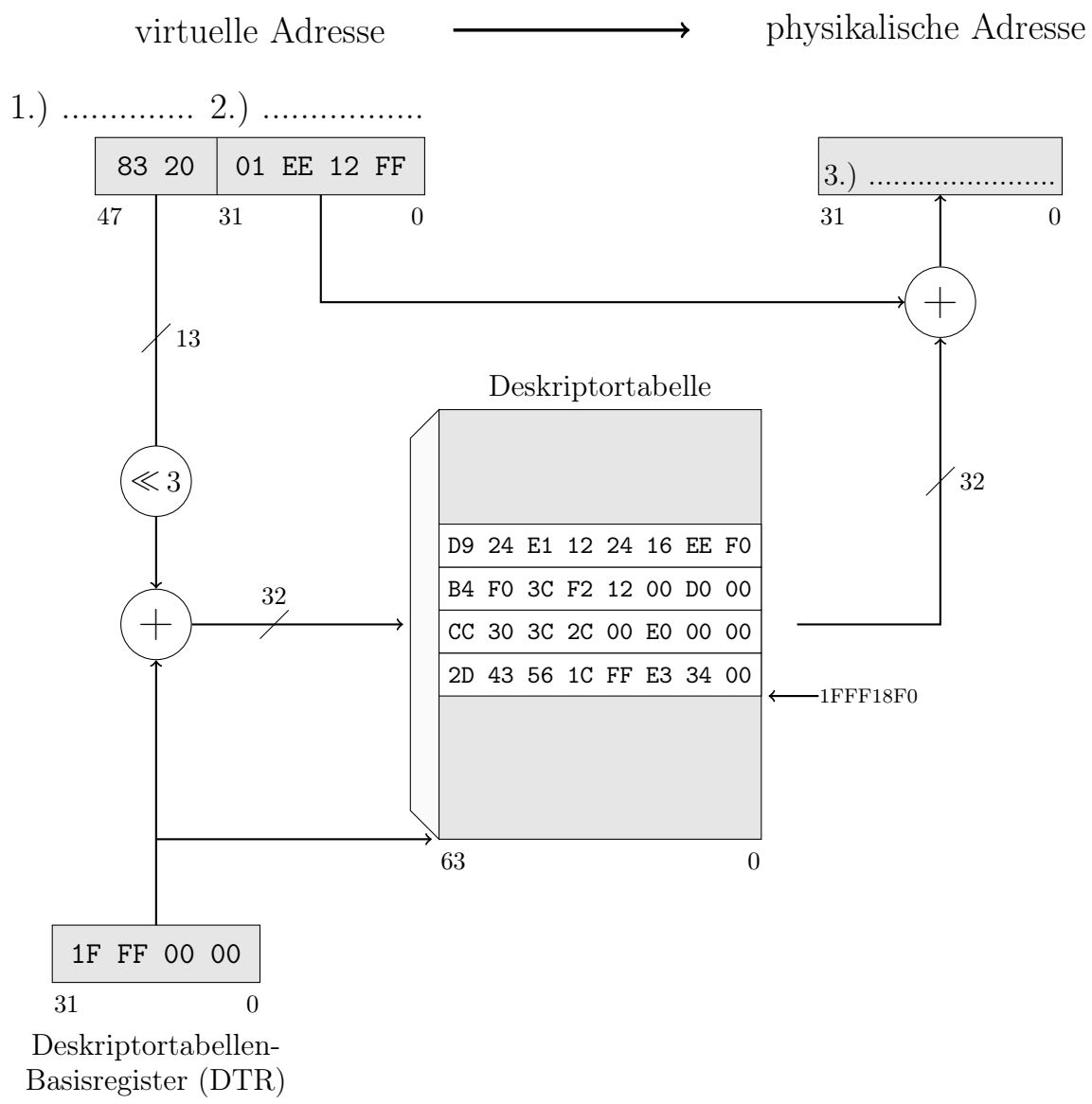
Z1:		0	.	0	0	1	0	1
Z2:	+	1	.	0	0	1	0	0
		1	.	0	1	0	0	1

Bei dem Ergebnis (1.01001) handelt es sich um eine normalisierte Mantisse. Das Endergebnis lautet als IEEE-754-Zahl in binärer Darstellung:

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
0	1	0	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Aufgabe II-3: Speicherverwaltung (20 P)

Gegeben sei ein 32-Bit-Prozessor mit Segmentierung und byteadressierbarem Speicher. Betrachten Sie dazu die unten aufgeführte schematische Abbildung. Registerbelegungen, Einträge im Speicher und Adressen sind im Hexadezimalformat angegeben und Register- bzw. Pfad-Bitbreiten im Dezimalformat. Die Bits 32 bis 44 der virtuellen Adresse werden dazu verwendet, zusammen mit der Deskriptortabellen-Basisadresse einen Segment-Deskriptor zu selektieren. Segment-Deskriptoren sind 64 Bit lang und enthalten die Segment-Basisadresse (Bits 0 bis 31) und Angaben zur Segmentlänge und zum Speicherschutz (Bits 32 bis 63). Die Segment-Basisadresse ergibt anschließend zusammen mit den Bits 0 bis 31 der virtuellen Adresse die physikalische Adresse.



a) Tragen Sie in der Abbildung die Bezeichner für die beiden Teile der virtuellen Adresse ein (1 und 2).

b) Geben Sie an, wie viele Segment-Deskriptoren die Deskriptortabelle maximal bei der in der Aufgabenstellung genannten Aufteilung der virtuellen Adresse haben kann und wie groß die Tabelle maximal wird (in KB).

Anzahl Segment-Deskriptoren: .....

Größe der Deskriptortabelle: .....

c) Geben Sie den Speicherbereich an, den die Segment-Deskriptortabelle belegt.

..... - .....

d) Geben Sie den zur Selektierung eines Deskriptors benötigten Teil der virtuellen Adresse vor und nach der Verschiebung ($\ll 3$) an.

Vor der Verschiebung: .....

Nach der Verschiebung: .....

e) Geben Sie die Startadresse des selektierten Segment-Deskriptors an.

.....

f) Geben Sie die enthaltene Segment-Basisadresse an.

.....

g) Berechnen Sie die physikalische Adresse und tragen Sie diese an der entsprechenden Stelle (3) in die Abbildung ein.

Lösungsvorschläge

a) Selektor (1) und Offset (2).

b) max. Anzahl: 13 Bit des Selektors zur Auswahl → 8192 mögliche Deskriptoren.
max. Größe: 8 Byte/Deskriptor · 8192 Deskriptoren = 64 KB oder 65536 Byte.

c) 1FFF0000 (Basisadresse) bis 1FFF0000 + 8*8192 -1 = 1FFFFFFF

d) Vor der Verschiebung: 320
Nach der Verschiebung: 1900

e) Startadresse des selektierten Segments: 1FFF1900

f) Segment-Basisadresse: 1200D000

g) physikalische Adresse (3): 1200D000 + 01EE12FF = 13EEE2FF

Aufgabe II-4: Sprungvorhersage (10 P)

Gegeben sei ein mehrmals durchlaufener Programmausschnitt, welcher die drei bedingten Sprünge B1, B2 und B3 enthält. Die folgende Tabelle gibt an, bei welchem Durchlauf jeweils ein Sprung genommen (T) oder nicht genommen (NT) wurden:

Durchlauf	B1	B2	B3
1	NT	T	T
2	T	NT	T
3	NT	T	T
4	NT	T	T
5	T	NT	NT

a) Für den Sprung B1 wird ein Ein-Bit-Prädiktor eingesetzt, welcher die zukünftige Sprungrichtung vorhersagen soll. Füllen Sie dazu die folgende Tabelle aus und geben Sie den Zustand des Prädikators bei jedem Durchlauf an (T – Taken, NT – Not Taken). Der Ein-Bit-Prädiktor wird zu Beginn mit NT initialisiert. Markieren Sie außerdem in der letzten Spalte falsche Vorhersagen.

Durchlauf	Vorhersage	Sprungauswertung	neue Vorhersage	Fehler
1				
2				
3				
4				
5				

b) Für den Sprung B2 wird ein Zwei-Bit-Prädiktor mit Sättigungszähler eingesetzt, welcher die zukünftige Sprungrichtung vorhersagen soll. Füllen Sie dazu die folgende Tabelle aus und geben Sie den Zustand des Prädikators bei jedem Durchlauf an (ST – Strongly Taken, SNT – Strongly Not Taken, WT – Weakly Taken, WNT – Weakly Not Taken). Der Zwei-Bit-Prädiktor wird zu Beginn mit WNT initialisiert. Markieren Sie außerdem in der letzten Spalte falsche Vorhersagen.

Durchlauf	Vorhersage	Sprungauswertung	neue Vorhersage	Fehler
1				
2				
3				
4				
5				

c) Welches Programmkonstrukt könnte hier aufgrund der Sprungverläufe durchlaufen werden?

. . . . . .

. . . . . .

d) Welche Sprünge würden sich für einen Korrelationsprädiktor besonders eignen und warum?

. . . . . .

. . . . . .

ENDE

Lösungsvorschläge

a) Lösung:

Durchlauf	Vorhersage	Sprungauswertung	neue Vorhersage	Fehler
1	NT	NT	NT	
2	NT	T	T	*
3	T	NT	NT	*
4	NT	NT	NT	
5	NT	T	T	*

b) Lösung

Durchlauf	Vorhersage	Sprungauswertung	neue Vorhersage	Fehler
1	WNT	T	WT	*
2	WT	NT	WNT	*
3	WNT	T	WT	*
4	WT	T	ST	
5	ST	NT	WT	*

c) Aufgrund von Sprung B3 könnte das Programmkonstrukt eine Schleife sein, welche fünf Mal durchlaufen wird.

d) Die Sprünge B1 und B2 korrelieren scheinbar voneinander und eignen sich damit besonders für einen Korrelationsprädiktor.