

WS 13/14Klausur: Rechnerarchitektur

Datum: 28.01.14

Prüfer: Prof. Dr. J. Neuschwander

Zeitdauer: 90 Min.

Name:

Matr.-Nr.:

- MUSTERLÖSUNG -Hinweise:

Beschriften Sie jedes zusätzlich abgegebene Lösungsblatt deutlich lesbar mit Ihrem Namen und Ihrer Matrikelnummer.

Es sind für diese Klausur als Hilfsmittel nur Taschenrechner, ein handschriftliches DIN-A4 Blatt als Formelsammlung sowie der Befehlssatz des Prozessors M68000 zugelassen.

Bei jeder angegebenen Lösung muss der Lösungsweg nachvollziehbar sein.

39

Maximal erreichbare Punktzahl der Klausur: 38,5 PunkteMindestpunktzahl zum Bestehen der Klausur: 17 Punkte

Aufgabe 1: 10 Punkte

Aufgabe 2: 4,5 Punkte

Aufgabe 3: 4 Punkte

Aufgabe 4: 6 Punkte

Aufgabe 5: 5,5 Punkte

Aufgabe 6: 4 Punkte

Aufgabe 7: 4,5 Punkte
5

1. Aufgabe: (10 P)

Gegeben sei folgende Programmsequenz für einen RISC-Prozessor:

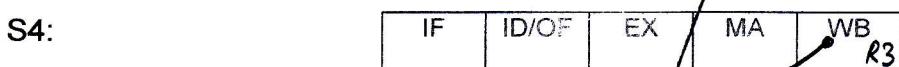
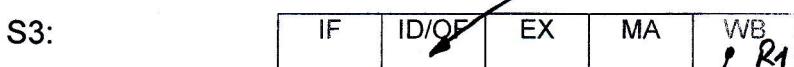
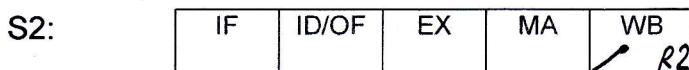
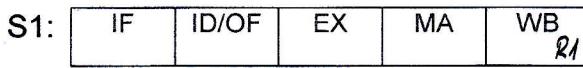
```

S1: LOAD R1, (A0) ; R1 := <A0>
S2: LOAD R2, (A1) ; R2 := <A1>
S3: ADD R1, R2, R3 ; R1 := R2 + R3
S4: MUL R3, R4, 5 ; R3 := R4 * 5
S5: ADD R3, R3, R1 ; R3 := R3 + R1
S6: STORE (A2), R3 ; <A2> := R3
S7: ADD R2, R1, R3 ; R2 := R1 + R3
    
```

Die einzelnen Befehle werden in einer DLX-Befehls-Pipeline verarbeitet. Die Latenzzeiten der einzelnen Stufen sind wie folgt:

IF	ID/OF	EX	MA	WB
4 ns	4 ns	3 ns	5 ns	3 ns

- a) Markieren Sie mit Pfeilen nur die Abhängigkeiten der Pipelinestufen, die im gegebenen Programm zu Pipeline-Konflikten führen.



$5 \times 0,4P = 2P$



- b) Wie viele Takte werden benötigt, um dieses Programm so abzuarbeiten?

$$\underline{T = k+n-1 = 5+7-1 = 11 \text{ Takte}}$$

0,5P

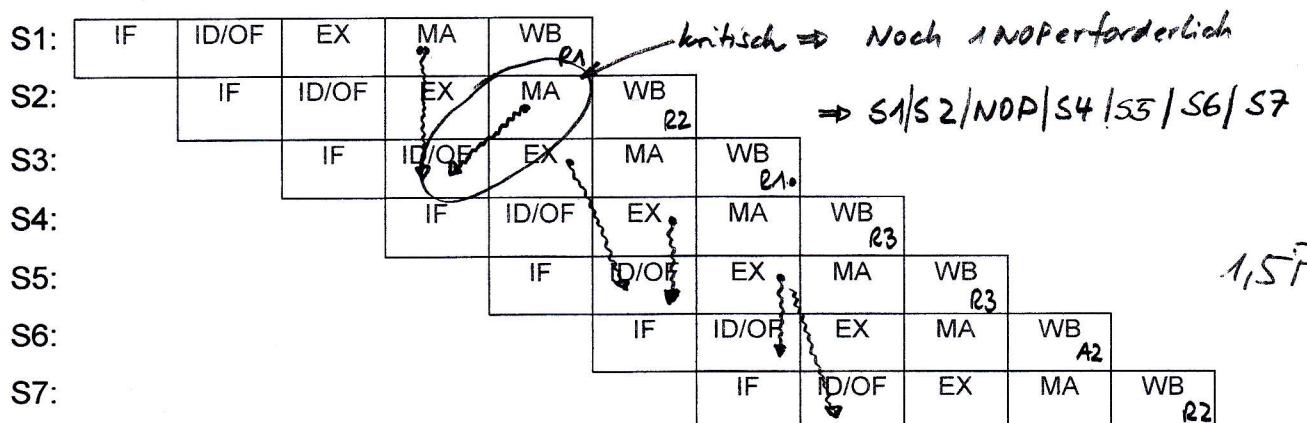
- c) Die auftretenden Pipeline-Konflikte sollen vom Compiler behandelt werden. Ergänzen Sie obiges Programmstück so, dass diese Pipeline-Konflikte beseitigt werden und das Ergebnis dem der normalen, sequentiellen Bearbeitung entspricht.

$S_1 / S_2 / NOP / NOP / S_3 / S_4 / NOP / NOP / S_5 / NOP / NOP / S_6 / S_7$

z also $6 \times NOP$ erforderlich

1,5P

- d) Wie viele NOP's sind im Programm noch erforderlich, wenn Load- und Result-Forwarding angewendet werden? Kennzeichnen Sie das jeweilige Forwarding.



- e) Die Latenzzzeiten der einzelnen Pipelinestufen sind am Anfang der Aufgabe angegeben. Die Latenzzeit eines Pipelineregisters beträgt 1,25 ns. Mit welcher maximalen Taktfrequenz kann die Pipeline betrieben werden?

$$\tau_{max} = \tau_{max\text{ Stufe}} + \tau_{Reg} = 5\text{ns} + 1,25\text{ns} = 6,25\text{ns}$$

$$f_{max} = \frac{1}{\tau_{max}} = \frac{1}{6,25\text{ns}} = \frac{1}{6,25} \cdot 10^9 \frac{1}{\text{s}} = 0,16 \cdot 10^9 \frac{1}{\text{s}} = 160 \text{MHz}$$

1P

- f) Wie viele Befehle muss ein Programm P mindestens besitzen, wenn der Speed-up der Pipeline den Wert 4,95 annehmen soll?

$$S = \frac{n \cdot k}{k+n-1} \Rightarrow S \cdot k + S \cdot n - S = n \cdot k \Rightarrow S \cdot n - n \cdot k = S - S \cdot k$$

$$n \cdot (S - k) = S(1 - k) \Rightarrow n \geq \frac{S(1 - k)}{S - k} \geq \frac{(1 - 5) \cdot 4,95}{4,95 - 5} = \underline{\underline{396}}$$

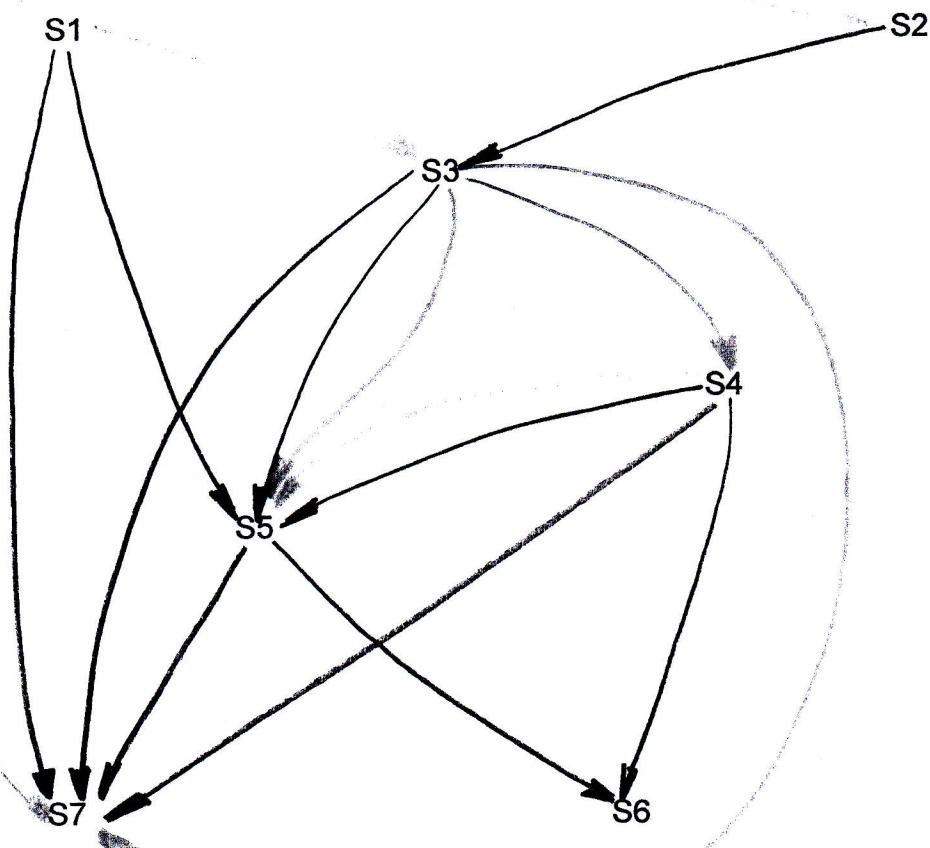
\Rightarrow P muss mindestens 396 Befehle enthalten.

1,5P

g) Geben Sie alle Datenabhängigkeiten im Programmstück an.

S1:	LOAD	R1, (A0)	; R1:= <A0>
S2:	LOAD	R2, (A1)	; R2:= <A1>
S3:	ADD	R1, R2, R3	; R1:= R2 + R3
S4:	MUL	R3, R4, 5	; R3:= R4 * 5
S5:	ADD	R3, R3, R1	; R3:= R3 + R1
S6:	STORE	(A2), R3	; <A2>:= R3
S7:	ADD	R2, R1, R3	; R2:= R1 + R3

Stellen Sie die jeweiligen Abhängigkeiten durch einen Pfeil dar und benennen Sie diese Abhängigkeit.



$$16 \times 0,125P = 2P$$

δt

δa

16 Abhängigkeiten

- $3 \times \delta^0$
- $3 \times \delta^a$
- $10 \times \delta^t$

2. Aufgabe: (4,5 P)

- a) Durch architekturelles Re-Design einer CPU wird die Anzahl der Taktzyklen für ein Programm P um 20% reduziert. Die Laufzeit von P soll jedoch insgesamt um 30% reduziert werden. Welche Taktfrequenz muss dazu gewählt werden, wenn die ursprüngliche Frequenz vor dem Re-Design 800 MHz betragen hat?

$$CPU_{Laufz} = \frac{n \cdot CPI}{f}$$

$$CPU_{Laufz} = \frac{n \cdot CPI_{vorher} \times 0,8}{f_{nachher}}$$

$$\frac{CPU_{Laufz}}{CPU_{Laufz}} = \frac{n \cdot CPI_{vorh} \times 0,8 \cdot f_{vorh}}{f_{nach} \cdot n \cdot CPI_{vorh}}$$

$$\frac{CPU_{Laufz} \times 0,7}{CPU_{Laufz}} = \frac{f_{vorh} \times 0,8}{f_{nach}} \Rightarrow f_{nach} = \frac{f_{vorh} \times 0,8}{0,7} = f_{vorh} \cdot \frac{0,8}{0,7}$$

$$\underline{\underline{f_{nachher}}} = 800 \text{ MHz} \cdot \frac{0,8}{0,7} = \underline{\underline{914,29 \text{ MHz}}}$$

2P

- b) Ein Rechnersystem wird durch zusätzliche Spezialhardware in der CPU verbessert. Durch diese CPU-Erweiterung laufen Berechnungen um den Faktor 20 schneller als bei herkömmlicher Ausführung. Verwenden Sie das Gesetz von Amdahl, um folgende Fragen zu beantworten:

- I. Wie viel Prozent des Codes müssen sich mit Hilfe der CPU-Erweiterung beschleunigen lassen, um einen Speedup (Gesamtbeschleunigung) von 3 zu erreichen?

$$S = \frac{1}{\overline{T}_{AU} + \frac{\overline{T}_{AB}}{B}} \Rightarrow S \cdot \overline{T}_{AU} + \overline{T}_{AB} \cdot \frac{S}{B} = 1 ; \quad \overline{T}_{AU} + \overline{T}_{AB} = 1$$

$$\overline{T}_{AB} = 1 - \overline{T}_{AU} \text{ oder} \\ \overline{T}_{AU} = 1 - \overline{T}_{AB}$$

$$\Rightarrow S \cdot (1 - \overline{T}_{AB}) + \overline{T}_{AB} \cdot \frac{S}{B} = 1$$

$$S - S \cdot \overline{T}_{AB} + \frac{S}{B} \overline{T}_{AB} = 1 \Rightarrow \overline{T}_{AB} \cdot \left(S - \frac{S}{B} \right) = S - 1$$

2P

$$\overline{T}_{AB} = \frac{S-1}{S - \frac{S}{B}} = \frac{2}{3 - \frac{3}{20}} = 0,7017 \Rightarrow \underline{\underline{\text{x 70% vom Code müssen auf der Spezial-HW laufen.}}}$$

- II. Welcher Prozentsatz des Codes muss auf der Spezialhardware ausgeführt werden, um lediglich ein Drittel des Speedups aus I.) zu erreichen?

$$\rightarrow S = \frac{3}{3} = 1$$

$$\overline{T}_{AB} = \frac{S-1}{S - \frac{S}{B}} = \frac{1-1}{1 - \frac{1}{20}} = 0$$

\rightarrow also gar kein Code, da $S=1$ ist!

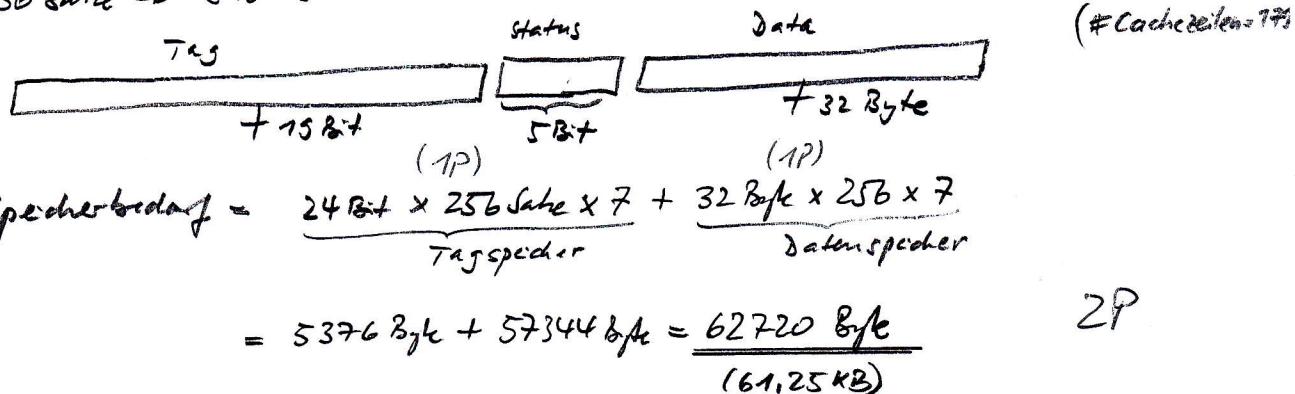
0,5P

3. Aufgabe: (4P)

3.1: Es soll ein 7-fach-satzassoziativer Cache-Speicher mit 256 Sätzen realisiert werden. Die Blockgröße im Hauptspeicher beträgt 32 Byte und die Hauptspeicheradresse ist 32 Bit breit. Zur Verwaltung eines Cache-Blocks werden 1 Valid-Bit, 1 Dirty-Bit und ein 3-Bit LRU-Zähler verwendet.

Bestimmen Sie den insgesamt erforderlichen Speicherbedarf für den Tag- und für den Datenspeicher zur Realisierung dieses Cache-Speichers.

Cachelinegröße 32 Byte \Rightarrow 5 Bit Byte-Navigation $\lceil 32 - (5+8) \rceil = 13$ Bit für Tag
256 Sätze \Rightarrow 8 Bit Satzindex



2P

3.2: Gegeben sei ein m-fach-satzassoziativer Cache-Speicher mit einer Cachelinegröße von 64 Byte. Die Hauptspeicheradresse umfasst 32 Bit. Der Satzindex ist 12 Bit breit. Die Kapazität des Cache beträgt 2 MB.

Cachelinegröße 64 Byte \Rightarrow 6 Bit Byte-Offset
Satzindex = 12 Bit

a) Wie viele Bit umfasst der Tag?

$$\underline{\text{Taggröße}} := 32 \text{ Bit} - (6 \text{ Bit} - 12 \text{ Bit}) = \underline{14 \text{ Bit}}$$

0,5P

b) Wie groß ist in diesem Fall die Assoziativität m?

$$\text{Speicherkapazität: } 64 \text{ Byte} \times 2^{12} \times m = 2 \text{ MB} \Rightarrow m = \frac{2 \text{ MB}}{64 \text{ Byte} \times 2^{12}}$$

$$m = \frac{2^{21}}{2^6 \cdot 2^{12}} = \frac{2^{21}}{2^{18}} = 2^3 = \underline{8} \quad 1P$$

(2¹⁵ Zeilen)

c) Wie viele Befehle der Länge 32 Bit können sich gleichzeitig im Cache-Speicher befinden?

Befehllänge 32 Bit \equiv 4 Byte

$$\rightarrow \text{Max pro Zeile: } \frac{64 \text{ Byte}}{4 \text{ Byte}} = 16$$

$$\underline{\text{max. gesamtzahl der Befehle: }} 16 \times 2^{12} \times 8 = 2^4 \times 2^{12} \times 2^3 = \underline{2^{19}}$$

(524288)
0,5P

4. Aufgabe: (6P)

(Zur Ermittlung der Punktzahl in dieser Teilaufgabe werden von den richtig angekreuzten Aussagen die falsch angekreuzten Aussagen abgezogen; Nicht angekreuzte Aussagen zählen nicht und gehen somit nicht in die Bewertung ein.)

je 0,5 P

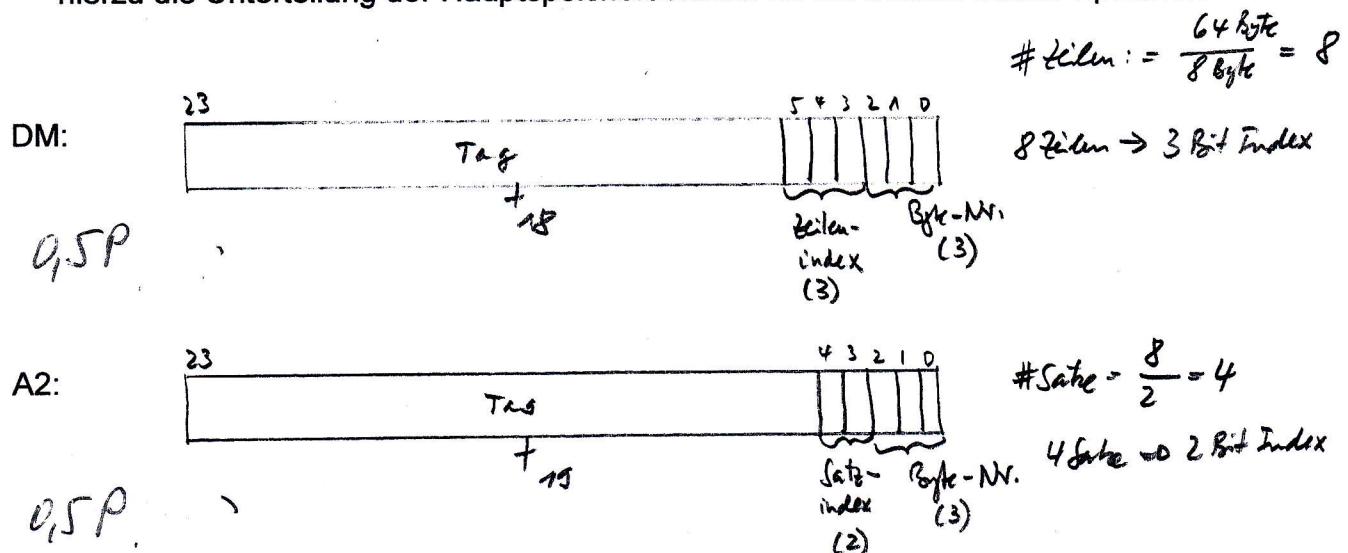
Fragen	richtig	falsch
Bei der Memory-Mapped-Adressierung liegen Speicheradressen und Ein-/Ausgabe-Adressen in separaten Adressräumen.		X
Mit Interleaving bezeichnet man die Verschachtelung von Hauptspeicherzugriffen auf verschiedene DRAM-Speicherbänke.	X	
Beim Simultaneous Multithreading (SMT) wird jedem Befehlsstrom ein eigener Registersatz zugewiesen.	X	
Die mittlere Zugriffszeit in einem Speichersystem, das aus einem Verbund von Cachespeicher und Arbeitsspeicher besteht, wird mit der Abnahme der Hitrate kleiner. $\tilde{T} = H \cdot T_{Cache} + (1-H) T_{Hauptspeicher}$		X
Alle Befehle innerhalb eines VLIW-Befehls müssen unabhängig voneinander sein und die Zuordnung der einzelnen Befehle zu den Ausführungseinheiten erfolgt dynamisch durch den Prozessor.		X
Bei SRAM-Speicherbausteinen kann die Zykluszeit durch Interleaving kompensiert werden.		X
Branch-Recovery dient zur Korrektur falscher Entscheidungen bei spekulativer Befehlsverarbeitung.	X	
Zur Lösung von Steuerflusskonflikten in einer Pipeline werden bei der verzögerten Sprungtechnik die Verzögerungszeiten durch NOP-Befehle ausgefüllt.	X	
Die History Bits im Branch-Target-Buffer (BTB) werden bei einer Pipeline jeweils in der Write-Back (WB)-Phase aktualisiert.	X	
Beim Cycle-by-Cycle-Interleaving (Multithreading) werden Threads solange ausgeführt, bis ein Befehl mit langer Latenzzeit auftritt.		X
Beim sogenannten Bus Snooping wertet der Prozessor die Adressen beim Zugriff anderer Busmaster auf den Hauptspeicher zur Konsistenzsicherung aus.		X
SIMD-Maschinen führen verschiedene Befehle auf denselben Daten aus (Array-Prozessoren).		X

5. Aufgabe: (5,5 P)

Gegeben seien ein *Direct-Mapped-Cache* (DM) und ein *2-Way-Set-Associative Cache* (A2). Die beiden Cache-Speicher haben jeweils eine Speicherkapazität von 64 Bytes und werden in Blöcken von je 8 Bytes geladen. Die Hauptspeicheradresse umfasst 24 Bit. Falls erforderlich, wird die „Least Recently Used“-Ersetzungsstrategie verwendet.

Als Aktualisierungsstrategie wird das *Copy-Back*-Verfahren verwendet.

- a) Welche Bits der 24-Bit Adresse bilden Offset, Tag und Index? Skizzieren Sie hierzu die Unterteilung der Hauptspeicheradresse für die beiden Cache-Speicher.



Betrachten Sie die folgenden Lese- und Schreibzugriffe auf die in hexadezimaler Schreibweise angegebenen Hauptspeicheradressen:

Adresse	\$8A	\$12	\$6C	\$9A	\$34	\$54	\$68	\$FE	\$17
read/write	r	r	w	w	r	w	w	r	r

Gehen Sie davon aus, dass zum Start dieser Adressfolge die beiden Cachespeicher leer sind.

- b) Geben Sie in der folgenden Tabelle an, ob es sich beim Zugriff auf die jeweiligen Adressen um einen Cache-Miss (Kennzeichnung „-“) oder einen Cache-Hit (Kennzeichnung „X“) handelt und geben Sie auch an, ob der entsprechende Cacheblock in den Hauptspeicher zurückkopiert werden muss (ja) oder nicht (nein).

Adresse	\$8A	\$12	\$6C	\$9A	\$34	\$54	\$68	\$FE	\$17
read/write	r	r	w	w	r	w	w	r	r
Hit / Miss	-	-	-	-	-	-	X	-	-
write back ?	n	n	n	n	n	n	n	n	j

Tabelle (falls benötigt) als Hilfe:

Hier nur DM-Cache zu betrachten!

Adresse	Binär	Index	Tag				Hit's
\$8A	10001010	1	2				
\$12	00010010	2	0			?	3
\$6C	01101100	5	1			6	0
\$9A	10011010	3	2			5	1v
\$34	00110100	6	0			4	
\$54	01010100	2	1			3	2
\$68	01101000	5	1			2	010
\$FE	11111110	7	3			1	2
\$17	00010111	2	0			0	

Index Byte-
Nr.

• Tabelle richtig (Struktur + Inhalt) = 3P

• Hit/Miss + write back-Tabelle OK = 1,5

6. Aufgabe: (4 P)

Der in einem 32-Bit-Rechnersystem eingesetzte *TLB* (*Translation Look Aside Buffer*) dient der Beschleunigung der Adressumsetzung. Der Adressspeicher des betrachteten TLB ist 24 Bit breit und teilt sich auf in eine Segmentnummer (8Bit) und eine Seitennummer (13 Bit). Der Datenspeicher, der die Framenummer enthält, ist 12 Bit breit.

- a) Skizzieren Sie die Unterteilung der virtuellen Adresse (jeweils Längenangaben in Bit).

31	Segm.-Nr.	Seiten-Nr.	Byte-Nr.	0
	8+	+13	+11	

0,5P

- b) Wie groß darf ein Segment des virtuellen Speichers maximal sein?

$$\underline{\text{Max. Segmentgröße}} = \frac{4\text{GB}}{2^8} = \frac{2^{32}\text{B}}{2^8} = 2^{24}\text{B} = \underline{16\text{MB}} \quad 0,5\text{P}$$

- c) Wie groß ist eine Seite und wieviele Seiten passen maximal in ein Segment?

$$\text{Seitengröße: } 2^n\text{B} = 2\text{kB}$$

$$\underline{\# \text{Seiten pro Segment}_{\max}} = \frac{16\text{MB}}{2\text{kB}} = \frac{2^{24}\text{B}}{2^{11}\text{B}} = \underline{2^{13}}$$

(oder oben aus a) ablesen)

0,5P

- d) Mit wieviel Byte muss der Hauptspeicher minimal ausgebaut sein, damit alle Frames darin Platz finden?

$$\underline{\text{Kapazität HS}} = \underline{\text{Framegröße}} \times \underline{\text{Anzahl Frames}} \quad 0,5\text{P}$$

$$= 2\text{kB} \times 2^{12} = 2^n\text{B} \times 2^{12} = 2^{23}\text{B} = \underline{8\text{MB}}$$

- e) Wie groß ist ein Programm P (in Byte), wenn es im Hauptspeicher 5 Frames belegt, wobei der letzte Frame nur zu 75% belegt ist?

$$\underline{\text{Programmgröße}} = \underbrace{4 \times 2\text{kB}}_{8\text{kB}} + \underbrace{0,75 \cdot 2\text{kB}}_{1,5\text{kB}} = \underline{9728\text{Byte (9,5kB)}}$$

0,5P

Gegeben sei nun eine andere Speicherverwaltungseinheit (MMU). Der virtuelle Speicher ist in 8 Seiten mit je 1 kByte unterteilt. Der physische Speicher hat eine Kapazität von 4 kByte. Der aktuelle Ausschnitt der Seitentabelle ist in folgender Tabelle angegeben.

Virtuelle Seitennummer	Physische Framenummer
0	3
1	1
2	-
3	-
4	2
5	-
6	0
7	-

$$1 \text{ kB} = 2^{10} \text{ Byte}$$

3 Adresse umfasst 10 Bit ($2^0 \dots 2^9$)

- f) Ermitteln Sie die physische Adresse (Dezimaldarstellung) zu den folgenden virtuellen Adressen (Dezimaldarstellung) :

$$4096 \stackrel{1}{=} 1000000000000 \quad (1)$$

↓

Byte-Nr.

virtuelle Seiten-Nr. 4 → Frame-Nr. 2

→ 010000000000 = 2048₁₀

- g) Ermitteln Sie die virtuelle Adresse (Dezimaldarstellung) zu den folgenden physischen Adressen (Dezimaldarstellung) :

$$4095 \stackrel{1}{=} 111111111111 \quad (0.5)$$

↓

Byte-Nr.

Frame-Nr. → Seiten-Nr = 0

⇒ 011111111111 = 1023₁₀

1,5P

5
7. Aufgabe: (4,5 P)

In der folgenden Aufgabe soll für alle Teilaufgaben jeweils die in Tabelle 8.1 dargestellte Ausgangssituation herrschen. Das Zeichen \$ kennzeichnet die hexadezimale Darstellung von Zahlen.

Prozessor-Register				
Datenregister	Inhalt			
D0	\$ 00	00	11	11
D1	\$ 22	22	33	33
D2	\$ 44	44	55	55
D3	\$ 66	66	77	77
D4	\$ 88	88	99	99
D5	\$ AA	AA	BB	BB
D6	\$ CC	CC	DD	DD
D7	\$ EE	EE	FF	FF

Arbeitsspeicher (16 Bit)	
Adresse	Inhalt
\$ 10000	\$ 00 01
\$ 10002	\$ 02 03
\$ 10004	\$ 04 05
\$ 10006	\$ 06 07
\$ 10008	\$ 08 09
\$ 1000A	\$ 0A 0B
\$ 1000C	\$ 0C 0D
\$ 1000E	\$ 0E 0F
\$ 10010	\$ 10 11
\$ 10012	\$ 12 13

Adressregister	Inhalt
A0	\$ 00 01 00 02
A1	\$ 00 01 00 06
A2	\$ 00 01 00 0A
A3	\$ 00 01 00 0E
A4	\$ 00 01 00 00
A5	\$ 00 01 00 00
A6	\$ 00 01 00 00
A7	\$ 00 0F FF FA

\$ FFFFA	\$ 01 02
\$ FFFFC	\$ 03 04
\$ FFFFE	\$ 05 06

Tabelle 8.1 : Ausgangssituation Register und Hauptspeicher

- a) Tragen Sie den Inhalt der Datenregister D1 und D4 (je 32 Bit) in Tabelle 8.2 ein, nachdem mit der Ausgangssituation entsprechend Tabelle 8.1 alle folgenden Befehle nacheinander ausgeführt wurden :

K1	DC.W	\$1234
K2	EQU	\$1234
ERG	DS.L	\$2
	MOVE.L	K1, D1
	ADD.W	#K2, D1
	SWAP	D1
	CMP.W	#0,D1
	BGT	M1
	SUB.W	D3, D4
	SWAP	D4
M1	MOVE.L	D4, ERG

$$\begin{array}{r}
 D1: 0000\ 1234 \\
 1234 + \\
 \hline
 D1: 00002468
 \end{array}$$

Prozessor-Register	
Datenregister	Inhalt
D1	\$ 2468 0000
D4	\$ 2222 8888

1P
0,5P

Tabelle 8.2 : Lösung zu Teilaufgabe a)

$$\begin{array}{r}
 D4: 88889999 \\
 D3: 66667777 \\
 \hline
 D4: 88882222
 \end{array} \quad (\text{SUB.W})$$

- b) Tragen Sie den Inhalt der Datenregister D0, D1 und D2 sowie der Adressregister A0 und A1 (je 32 Bit) in Tabelle 8.3 ein, nachdem mit der Ausgangssituation entsprechend Tabelle 8.1 die folgenden Befehle nacheinander ausgeführt wurden:

```

ANF EQU      $FFFFC
MOVE.L #ANF, A0
MOVE.W (A2)+, D0
MOVE.L (A2)+, D1

```

Prozessor-Register	
Datenregister	Inhalt
D0	\$ 00000A0B
D1	\$ 0C0D0E0F

Adressregister	
Adressregister	Inhalt
A0	\$ 000FFFFC

Tabelle 8.3 : Lösung zu Teilaufgabe b)

- c) Tragen Sie den Inhalt der Datenregister D3 und D4 (je 32 Bit) in Tabelle 8.4 ein, nachdem mit der Ausgangssituation entsprechend Tabelle 8.1 die folgenden Befehle nacheinander ausgeführt wurden:

```

SWAP D3
ADD.L (A0)+, D3
LSL.W #2, D4

+ 77 77 66 66
  02 03 04 05
  _____
  79 7A 6A 6B

```

Prozessor-Register	
Datenregister	Inhalt
D3	\$ 797A6A6B
D4	\$ 88886664

Tabelle 8.4 : Lösung zu Teilaufgabe c)

84: 9 9 9 9 ← LSL

1001100110011001

0110011001100100

 6 6 6 4