

WS 12/13Klausur: Rechnerarchitektur

Datum: 12.02.13

Prüfer: Prof. Dr. J. Neuschwander

Zeitdauer: 90 Min.

Name:Matr.-Nr.:Hinweise:

Beschriften Sie jedes zusätzliche 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 erkennbar sein.

Maximal erreichbare Punktzahl der Klausur: 37 Punkte

Mindestpunktzahl zum Bestehen der Klausur: 17 Punkte

Aufgabe 1: 9,5 Punkte
Aufgabe 2: 4 Punkte
Aufgabe 3: 6 Punkte
Aufgabe 4: 6,5 Punkte
Aufgabe 5: 5 Punkte
Aufgabe 6: 6 Punkte

1. Aufgabe: (9,5 P)

Gegeben sei folgende Programmsequenz für einen RISC-Prozessor:

```

S1:  ADD    R0, R1, 7      ; R0:= R1 + 7
S2:  LOAD   R1, (A1)       ; R1:= <(A1)>
S3:  SUB    R3, R2, R0     ; R3:= R2 - R0
S4:  MUL    R2, R3, 4      ; R2:= R3 * 4
S5:  LOAD   R4, (A2)       ; R4:= <(A2)>
S6:  ADD    R2, R2, R0     ; R2:= R2 + R0
S7:  ADD    R1, R3, R4     ; R1:= R3 + R4
S8:  STORE  (A2), R4       ; <(A2)>:= R4

```

Die einzelnen Befehle werden in einer DLX-Befehls-Pipeline verarbeitet. Die Latenzzeiten der einzelnen Stufen sind in der Abbildung ersichtlich.

4 ns	5 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.

S1:

IF	ID/OF	EX	MA	WB
----	-------	----	----	----

S2:

IF	ID/OF	EX	MA	WB
----	-------	----	----	----

S3:

IF	ID/OF	EX	MA	WB
----	-------	----	----	----

S4:

IF	ID/OF	EX	MA	WB
----	-------	----	----	----

S5:

IF	ID/OF	EX	MA	WB
----	-------	----	----	----

S6:

IF	ID/OF	EX	MA	WB
----	-------	----	----	----

S7:

IF	ID/OF	EX	MA	WB
----	-------	----	----	----

S8:

IF	ID/OF	EX	MA	WB
----	-------	----	----	----

- b) 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.
- c) Wie viele Takte werden benötigt, um dieses Programm aus b) dann abzuarbeiten?
- d) Die Latenzzeiten der einzelnen Pipelinestufen sind am Anfang der Aufgabe angegeben. Die Latenzzeit der Pipelineregister beträgt 3 ns. Mit welcher maximalen Taktfrequenz (in MHz) kann die Pipeline betrieben werden?
- e) Welchen Wert hat der „Speed-up“ dieser Pipeline, wenn im Mittel 4000 Befehle pro Programm verarbeitet werden?
- f) Ein Rechnerarchitekt will den „Speed-up“ der Pipeline auf 7 erhöhen. Wie müsste die Stufenzahl der obigen Pipeline geändert werden, wenn im Mittel 4000 Befehle pro Programm verarbeitet werden?

g) Geben Sie alle Datenabhängigkeiten im Programmstück an.

S1:	ADD	R0, R1, 7	; R0:= R1 + 7
S2:	LOAD	R1, (A1)	; R1:= <(A1)>
S3:	SUB	R3, R2, R0	; R3:= R2 – R0
S4:	MUL	R2, R3, 4	; R2:= R3 * 4
S5:	LOAD	R4, (A2)	; R4:= <(A2)>
S6:	ADD	R2, R2, R0	; R2:= R2 + R0
S7:	ADD	R1, R3, R4	; R1:= R3 + R4
S8:	STORE	(A2), R4	; <(A2)>:= R4

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

S1

S2

S7

S3

S4

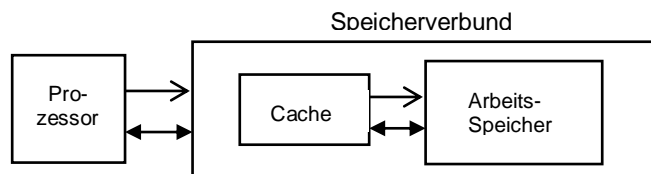
S6

S5

S8

2. Aufgabe: (4 P)

Der Prozessor in einem Rechnersystem arbeitet auf einem Speichersystem, das aus einem Verbund aus Cachespeicher und Arbeitsspeicher besteht (siehe Bild). Der Zugriff auf den Cachespeicher erfolgt innerhalb von 4 ns. Die mittlere Zugriffszeit auf das Verbundsystem beträgt 10 ns. Die Hitrate wird mit 80% angegeben.



- a) Wie lange dauert in diesem Fall die Zugriffszeit auf den Arbeitsspeicher?
- b) Wie würde sich die mittlere Zugriffszeit ändern, wenn die Hitrate auf 70% fällt?
- c) Gegeben sei das folgende Codefragment, dass auf einer superskalaren Pipeline ausgeführt werden soll.

$$\begin{aligned} Z &= A + B \\ R_1 &= C + Z \\ Z &= D + E \\ R_2 &= Z - F \end{aligned}$$

Beseitigen Sie alle durch WAR (bedingt durch δ^a) und WAW (bedingt durch δ^o) auftretenden Datenkonflikte mittels *Register-Renaming* und geben Sie eine mögliche Parallelisierung des modifizierten Codes an.

3. Aufgabe: (6,5 P)

Gegeben sei ein 2-fach-satzassoziativer Cache-Speicher (A2) mit einer Cachelinegröße von 32 Byte. Die Hauptspeicheradresse umfasst 32 Bit. Der Satzindex ist 13 Bit breit. Zur Verwaltung eines Cache-Blocks werden das Valid-Bit (*V*) und das Dirty-Bit (*D*) sowie ein Bit für die Alterungsinformation (LRU) verwendet.

- a) Wie viele Bit umfasst der Tag?

- b) Wie viele Befehle mit der Größe 32 Bit können sich gleichzeitig im Cache-Speicher befinden?

- c) Bestimmen Sie den insgesamt erforderlichen Speicherbedarf für die Realisierung des Cache-Speichers (Adress- und Datenspeicher).

Gegeben sei ein direkt abbildender Cache-Speicher mit einer Speicherkapazität von 128 Byte und einer Blockgröße von 16 Byte. Als Aktualisierungsstrategie wird das Rückschreibverfahren (*write back*) verwendet. Nehmen Sie an, dass der Cache-Speicher zu Beginn leer ist. Betrachten Sie die folgenden Lese- und Schreibzugriffe auf die in hexadezimaler Schreibweise angegebenen Adressen.

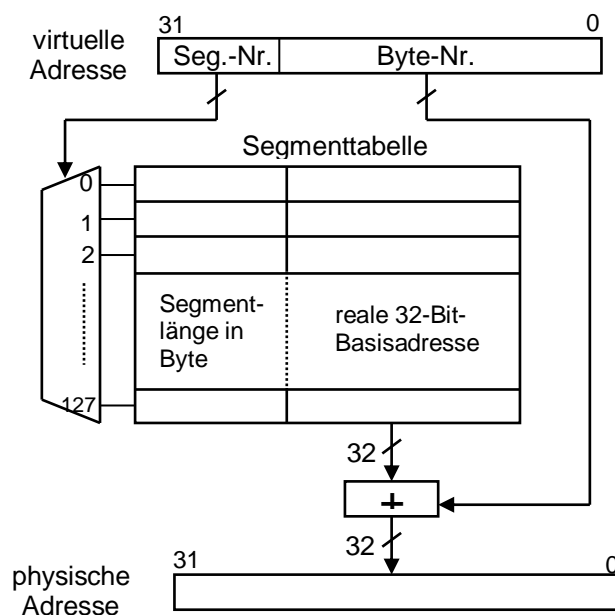
- d) Ergänzen Sie in dieser Tabelle die fehlenden Einträge für Index, Tag sowie für Hit/Miss und write back. Verwenden Sie dabei **M** für Cache-Miss und **H** für Cache-Hit. Geben Sie in der letzten Zeile der Tabelle an, ob der entsprechende Cache-Block in den Hauptspeicher zurückkopiert werden muss (**j**) oder nicht (**n**).

Adresse (hexadez.)	99	3A	96	25	15	D2	29	34	9A
read / write	r	w	w	r	r	w	w	w	w
Index									
Tag									
Hit / Miss									
write back ?									

4. Aufgabe: (5 P)

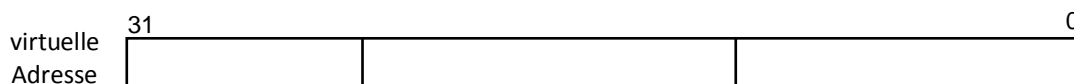
In einem Rechnersystem unterstützt eine Speicherverwaltungseinheit (MMU) eine segmentorientierte Speicherverwaltung zur Abbildung des virtuellen Adressraums auf den physischen Adressraum. Die Wirkungsweise dieser MMU ist im folgenden Bild dargestellt.

- a) Geben Sie die maximal mögliche Segmentgröße in MByte an.

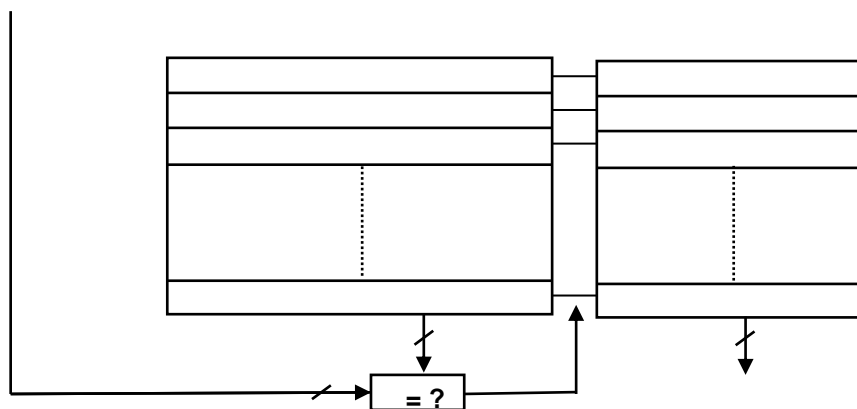
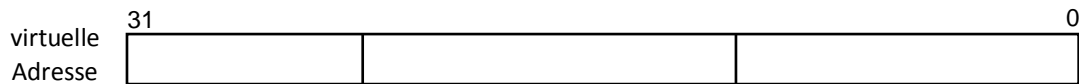


Zur Optimierung der Austauschstrategie bei der virtuellen Adressierung soll zusätzlich zu a) ein Seitenverfahren eingeführt werden. Der Hauptspeicherausbau umfasst die Hälfte des virtuellen Speichers, wobei für die Framegröße 4 KB festgelegt wurde.

- b) Geben Sie dazu die Unterteilung der virtuellen Adresse mit den einzelnen Feldlängen an. Kennzeichnen Sie die Felder, die von der Adreßumsetzung (virtuell → physikalisch) betroffen sind.



- c) Entwerfen Sie nun einen Adressumsetzungspuffer (TLB), der zur Beschleunigung der unter b) eingeführten Adressumsetzung eingesetzt werden soll. Er soll vom Typ Direct Mapped (DM) sein und 32 Zeilen besitzen. Ergänzen sie die untenstehende Skizze des TLB mit allen erforderlichen Merkmalen (Adressierung des TLB, Länge des Adresstags, Länge des Datenfelds, Größe der zu vergleichenden Tags).



- d) Auf welche dezimale physische Adresse wird bei der Konfiguration nach c) die dezimale virtuelle Adresse 8000 abgebildet? Die Abbildung zeigt einen Ausschnitt aus der Umsetzungstabelle der MMU.

Segment-nummer	Virtuelle Seitennummer	Physische Seitennummer
0	0	1
0	1	3
0	2	0
usw.	usw.	usw.

5. Aufgabe: (6 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

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

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

.

.

.

\$ 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 DC.W \$3333
 ERG DS.L \$2
 MOVE.L K1, D1
 ADD.W K2, D1
 SWAP D1
 CMP.W #0,D1
 BGT M1
 SUB.L D3, D4
 M1 SUB.L D2, D4
 MOVE.L D4,ERG

Prozessor-Register	
Datenregister	Inhalt
D1	\$
D4	\$

Tabelle 8.2 : Lösung zu Teilaufgabe a)

- 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    $1000E
      MOVE.L #ANF, A0
      MOVEA.L ANF, A1
      MOVE.W (A2)+, D0
      MOVE.L (A2)+, D1
      MOVE.W -(A0), D2
```

Prozessor-Register	
Datenregister	Inhalt
D0	\$
D1	\$
D2	\$

Adressregister	
A0	Inhalt
A0	\$
A1	\$

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
LSR.W #2, D4
```

Prozessor-Register	
Datenregister	Inhalt
D3	\$
D4	\$

Tabelle 8.4 : Lösung zu Teilaufgabe c)

6. Aufgabe: (6 P)

(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.)

Fragen	richtig	falsch
Bei der Memory-Mapped-I/O sind die Register der Interface-Bausteine Teil eines separaten I/O-Adressraums des Prozessors.		
Unter Data-Alignment versteht man die Verschachtelung von Hauptspeicherzugriffen auf verschiedene Speicherbänke.		
Beim Simultaneous Multithreading (SMT) wird jedem Befehlsstrom ein eigener Registersatz zugewiesen.		
Beim asynchronen Systembus wird die Anzahl der Wartezyklen des Prozessors durch das READY-Signal gesteuert.		
Alle Befehle innerhalb eines VLIW-Befehls müssen unabhängig voneinander sein und die Zuordnung der einzelnen Befehle zu den Ausführungseinheiten erfolgt dynamisch durch den Prozessor.		
Bei SRAM-Speicherbausteinen kann die Zykluszeit durch Interleaving kompensiert werden.		
Branch-Recovery dient zur Korrektur falscher Entscheidungen bei spekulativer Befehlsverarbeitung.		
Zur Lösung von Steuerflusskonflikten in einer Pipeline werden bei der verzögerten Sprungtechnik Verzögerungszeiten durch NOP-Befehle kompensiert.		
Die Branch-History Bits im Branch-Target-Buffer (BTB) werden in der Execution Phase (EX) der Pipeline aktualisiert.		
Beim Cycle-by-Cycle-Interleaving (Multithreading) werden Threads solange ausgeführt, bis ein Befehl mit langer Latenzzeit auftritt.		
Beim sogenannten Bus Snooping wertet der Prozessor die Adressen beim Zugriff anderer Busmaster auf den Hauptspeicher zur Konsistenzsicherung aus.		
SIMD-Maschinen führen verschiedene Befehle auf denselben Daten aus (Array-Prozessoren).		