Klausur zur Vorlesung

Grundlagen der Rechnerarchitektur / Technische Informatik (GRA/TI)

Prof. Marco Platzner
Fachgebiet Technische Informatik
Universität Paderborn

25.09.2013

- Die Bearbeitungsdauer beträgt für alle Studenten **90 Minuten.** Es sind **alle 5 Aufgaben** zu bearbeiten.
- Es sind keine Hilfsmittel zugelassen.
- Schreiben Sie nicht mit Bleistift oder Rotstift.
- Verwenden Sie kein eigenes Papier. Bei Bedarf bekommen Sie Papier bei der Klausuraufsicht.
- Schreiben Sie auf jedes Blatt (auch auf das Konzeptpapier) in Blockschrift Ihren Namen und Ihre Matrikelnummer.
- Bei mehreren präsentierten Lösungen wird die Aufgabe nicht gewertet! Streichen Sie daher bei Angabe mehrerer Lösungsansätze die nicht zu bewertenden Lösungen durch! Verwenden Sie kein Tipp-Ex.
- Abschreiben und abschreiben lassen oder Hilfe Dritter führt zum Nichtbestehen der Klausur.

Nachname: _____

Vorname: _____

Matrikelnummer: _____

Studiengang: _____

Aufkleber

Aufgabe	**1**	**2**	**3**	**4**	**5**	$\sum$
Punkte	**15**	**20**	**15**	**20**	**20**	90
Erreicht						

Aufgabe 1 (Multiple Choice) **[15 Punkte]**

Bei den folgenden Fragen können keine, eine oder mehrere Antworten richtig sein. Kreuzen Sie die richtigen Antworten deutlich an.

(a) Was bedeutet die Big-Endian Konvention?

- ☐ Der Speicher wird mit Wortadressen adressiert.
- ☐ Das höchstwertigste Byte eines Wortes befindet sich an der niedrigsten Byte-Adresse.
- ☐ Das höchstwertigste Byte eines Wortes befindet sich an der höchsten Byte-Adresse.
- ☐ Der Prozessor ist geeignet für die Verarbeitung von großen Datenmengen.

(b) Was sind Charakteristika einer reinen Akkumulator-Maschine?

- ☐ Es gibt nur ein Register für arithmetische/logische Operationen.
- ☐ Maximal ein Operand befindet sich in einem Register.
- ☐ Operationen werden auf einem Stack durchgeführt.

(c) Welche Faktoren der Prozessorperformance hängen von der Technologie ab?

- ☐ I_c
- ☐ CPI
- ☐ T

(d) Welche Funktion hat das reference bit in der page table?

- ☐ Das reference bit dient zur Approximation der least recently used Ersetzungsstrategie.
- ☐ Das reference bit gibt an, ob die Seite vor dem Ersetzen in den Sekundärspeicher zurückkopiert werden muss.
- ☐ Das reference bit gibt an, ob sich der page table Eintrag auch im TLB befindet.
- ☐ Das reference bit gibt an, ob der page table Eintrag gültig ist.

(e) Welches Arbitrierungsschema ordnet die Einheiten in einer Prioritätskette an?

- ☐ Verteilte Arbitrierung durch Kollisionserkennung
- ☐ Daisy-Chain Arbitrierung
- ☐ Zentrale Arbitierung
- ☐ Verteilte Arbitrierung durch Selbstselektion

Aufgabe 2 (Assembler) **[20 Punkte]**

Nachfolgend ist ein MIPS Assemblerprogramm abgebildet. Die erste Spalte zeigt dabei die Adressen im Hexadezimalsystem, an denen die jeweiligen Instuktionen im Programmspeicher stehen.

```
0100  L0:  addi    $sp,     ______, _____   # Lege den Inhalt von
0104       sw      $ra,     ______          # $ra und $a0
0108       sw      $a0,     ______          # auf den Stack
010C       slti    $t0,     $a0,      1
0110       ______  $t0,     $zero,    L1    # Sprung zu L1, falls $a0 < 1
0114       addi    $v0,     $zero,    1
0118       addi    $sp,     ______,   _____ # Stack Pointer aktualisieren
011C       jr      ______                   # Ruecksprung
0120  L1:  addi    $a0,     $a0,      -1
0124       jal     L0
0128       lw      $a0,     ______          # $a0 und $ra vom
012C       lw      $ra,     ______          # Stack holen
0130       mul     $v0,     $a0,      $v0
0134       addi    $sp,     ______,   _____ # Stack Pointer aktualisieren
0138       jr      ______                   # Ruecksprung
```

(a) Vervollständigen sie das oben abgebildete Assemblerprogramm gemäß der im Code angegeben Kommentare.

(b) Analysieren Sie eine Ausführung des Assemblerprogramms mit der folgenden initialen Belegung der Register `$s0`, `$a0`, `$v0` und `$ra`:

Register	Wert
`$s0`	1000_{hex}
`$a0`	3
`$v0`	0
`$ra`	0010_{hex}

Geben Sie in der untenstehenden Tabelle die Registerbelegungen zu den ersten 5 Zeitpunkten des Erreichens der ersten Programmzeile an. Geben sie jeweils den Registerinhalt vor der Ausführung der Programmzeile an. Sollte die erste Zeile keine 5 mal erreicht werden, lassen sie die verbleibenden Spalten leer. **Hinweis:** *Lesen sie vor der Bearbeitung Aufgabenteil c.*

	1	2	3	4	5
`$s0`					
`$a0`					
`$v0`					
`$ra`					

(c) Geben sie zu den selben Zeitpunkten wie in Aufgabenteil b, die Belegungen der Speicherstellen in der untenstehenden Tabelle an. Tragen sie nur an den Speicherstellen Werte ein, an denen sie durch die Programmausführung bekannt sind. Lassen Sie alle anderen Felder leer.

	1	2	3	4	5
1008_{hex}					
1004_{hex}					
1000_{hex}					
$0FFC_{\text{hex}}$					
$0FF8_{\text{hex}}$					
$0FF4_{\text{hex}}$					
$0FF0_{\text{hex}}$					
$0FEC_{\text{hex}}$					
$0FE8_{\text{hex}}$					
$0FE4_{\text{hex}}$					
$0FE0_{\text{hex}}$					
$0FDC_{\text{hex}}$					
$0FD8_{\text{hex}}$					

(d) Welchen Wert hält das Register `$v0` nach Abarbeitung des Assemblerprogramms bei einer initialen Belegung von `$a0` mit dem Wert 4?

Aufgabe 3 (Performancevergleich) **[15 Punkte]**

Eine 5-stufige Pipelineimplementierung für die MIPS Architektur soll mit einer Mehrzyklenimplementierung anhand des folgenden Instruktionsmixes verglichen werden:

Instruktionsklasse	*relative Häufigkeit*	*CPI-Wert der Mehrzyklenimplementierung*
lw	24%	5
sw	12%	4
R-Type	40%	4
beq	15%	3
j	9%	3

Folgende kombinatorische Verzögerungszeiten sind gegeben: Speicherzugriff 200 ps, ALU-Operation bzw. eine Adder-Operation 100 ps, und Zugriff auf das Registerfile 50 ps. Alle anderen Verzögerungszeiten (z.B. der Multiplexer, Kontroller, Leitungen, Pipeliningregister, etc.) werden vernachlässigt.

(a) Geben Sie die Taktperiode beider Implementierungen an.

(b) Berechnen Sie den CPI-Wert der Mehrzyklenimplementierung.

(c) Wie groß ist der maximale Speedup der Pipelineimplementierung gegenüber der Mehrzyklenimplementierung im Idealfall (ohne Konflikte)?

(d) Für die Pipelineimplementierung wird nun angenommen:

- Drei-viertel aller load Instruktionen führen zu einem load-use Konflikt, der jeweils zu einer Verzögerung von einem Taktzyklus führt.
- 40% aller bedingten Sprunginstruktionen werden falsch vorhergesagt und die branch penalty beträgt einen Taktzyklus.
- Unbedingte Sprünge führen immer zu einem Pipeline Stall von einem Taktzyklus.
- Weitere mögliche Konflikte werden vernachlässigt.

Berechnen Sie den resultierenden maximalen Speedup der Pipelineimplementierung gegenüber der Mehrzyklenimplementierung für den angegebenen Instruktionsmix.

Aufgabe 4 (Pipelining) **[20 Punkte]**

Es sei ein Prozessor einer 5-stufigen Pipeline (IF, ID, EX, ME, WB) gegeben. Der Registerzugriff erfolgt im Ganztaktverfahren. Es steht ein gemeinsamer Speicher für Daten und Instruktionen zur Verfügung. Die Pipeline verwendet kein Forwarding.

Folgendes Assemblerprogramm soll auf dem Prozessor ausgeführt werden:

```
1:  lw   $1,  100($3)
2:  sub  $2,  $4,  $1
3:  add  $5,  $2,  $1
4:  mul  $6,  $2,  $1
5:  or   $7,  $2,  $1
6:  add  $8,  $2,  $1
7:  sw   $8,  100($3)
```

(a) Lösen Sie die Konflikte, indem die Pipeline entsprechend angehalten wird. Erstellen Sie ein Diagramm, aus dem die Pipelinebelegung ersichtlich wird.

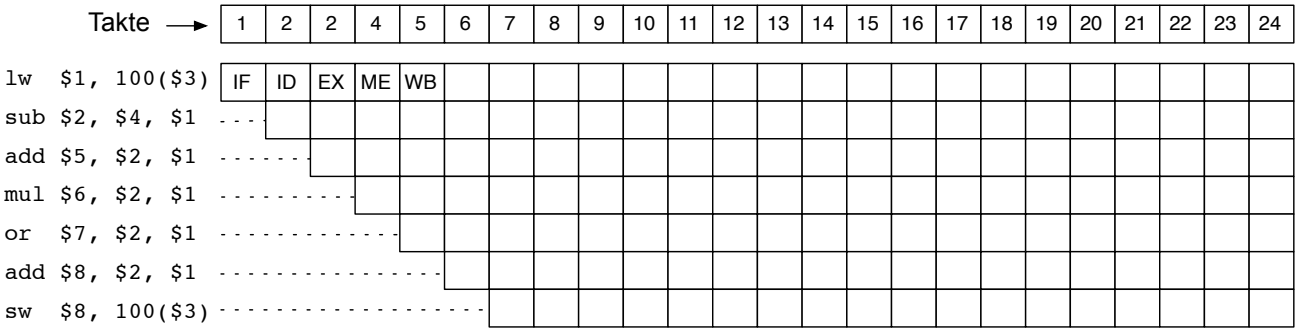

Abbildung 1: Pipelinebelegung zu (a)

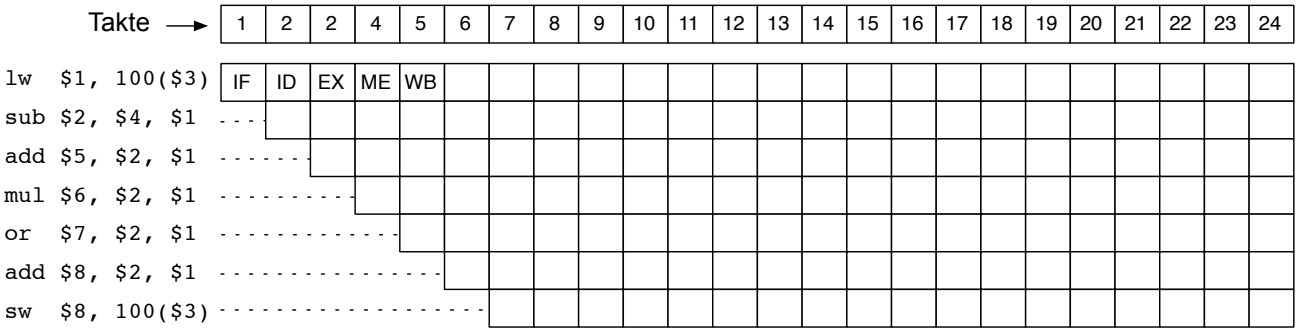

Abbildung 2: (Ersatzdiagramm) Pipelinebelegung zu (a)

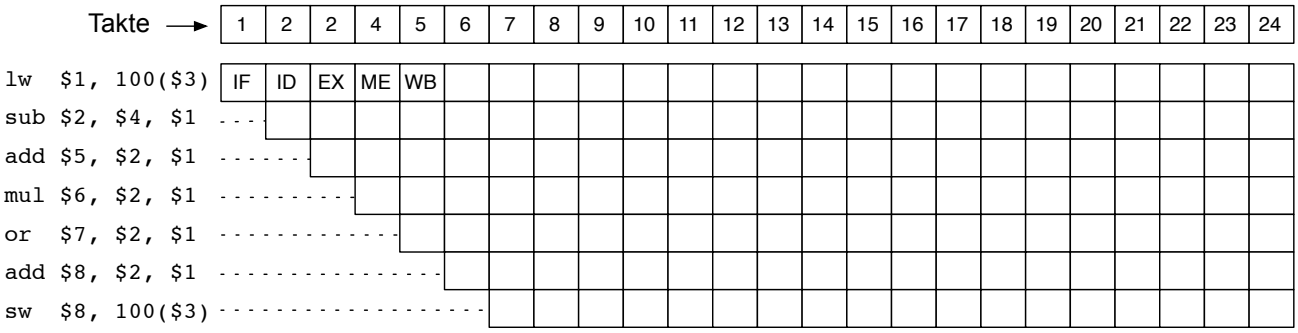

Abbildung 3: (Ersatzdiagramm) Pipelinebelegung zu (a)

(b) Gehen Sie nun von einem Prozessor aus, der zusätzlich Forwarding und einen Registerzugriff im Halbtaktverfahren unterstützt. Im Fall eines durch das Forwarding nicht auflösbaren Konflikts wird die Pipeline angehalten. Erstellen Sie ein Diagramm, aus dem die Pipelinebelegung für diesen erweiterten Prozessor bei dem gegebenen Assemblerprogramm ersichtlich wird. Machen Sie das Forwarding mit einem Pfeil kenntlich.

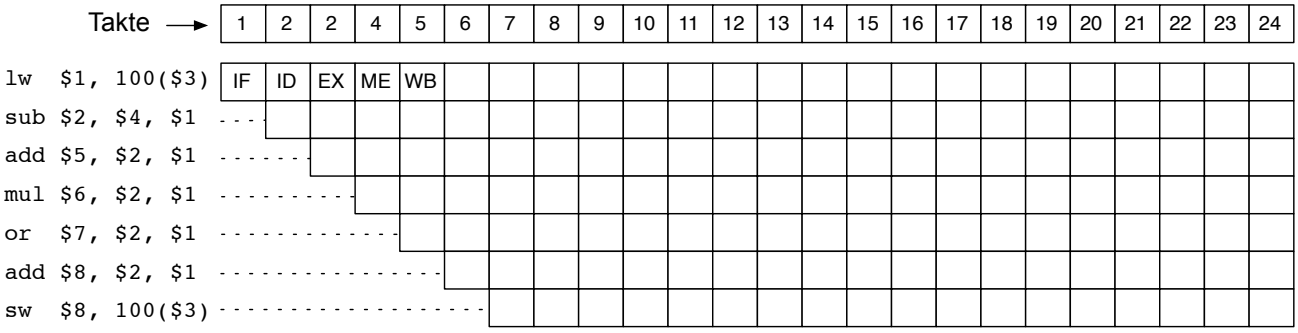

Abbildung 4: Pipelinebelegung zu (b)

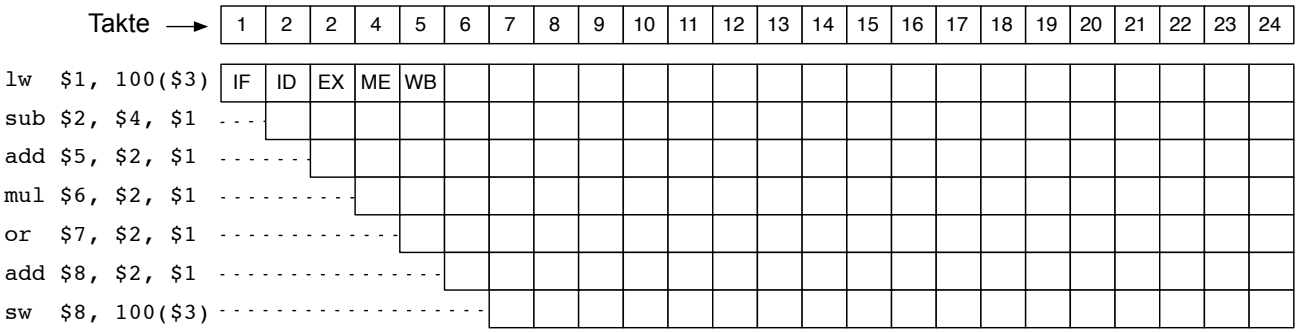

Abbildung 5: (Ersatzdiagramm) Pipelinebelegung zu (b)

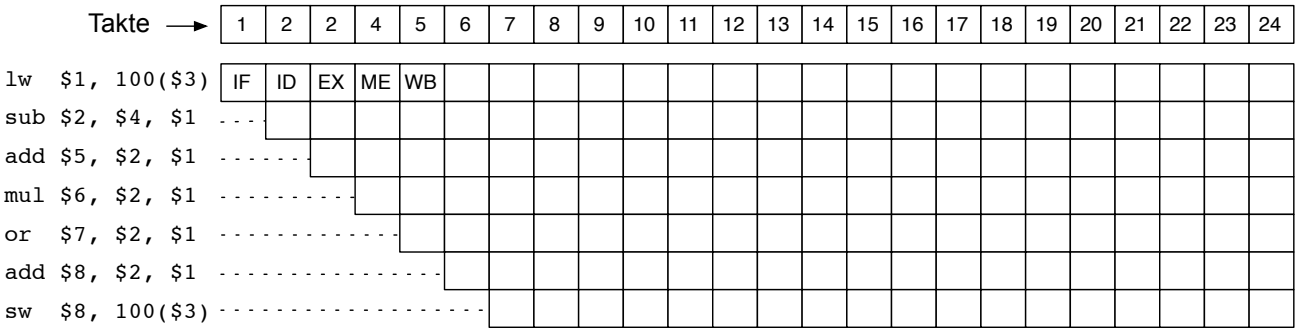

Abbildung 6: (Ersatzdiagramm) Pipelinebelegung zu (b)

(c) Wird eine weitere Optimierung des Prozessors um eine Harvard-Architektur hier in einer kürzeren Ausführungszeit resultieren? Begründen Sie Ihre Antwort. Falls ja, berechnen Sie den Speedup dieses neuen Prozessors gegenüber dem Prozessor aus Teilaufgabe (b).

(d) Wie wird Forwarding in Hardware realisiert? Nennen Sie die wichtigsten Vor- und Nachteile.

Aufgabe 5 (Cache) **[20 Punkte]**

Gehen Sie von einem Computer aus, der zur Beschleunigung der Speicherzugriffe auf den byteadressierten Arbeitsspeicher mit einer Kapazität von 2 GB einen Cache mit 8 Rahmen vorsieht, wobei jeder Rahmen 4 Worte zu je 32 Bit umfasst. Der Cache sei als direkt abgebildeter Cache organisiert.

(a) Geben Sie die Breite des Adressbus sowie die Anzahl an Bits für Tag, Index und Offsets an.

(3 Punkte)

Adressbus: _____

Tag: _____

Index: _____

Wortoffset: _____

Byteoffset: _____

(b) Wie viel Speicher wird für die Implementierung eines solchen Cache (inkl. Organisation) benötigt? Geben Sie den Rechenweg und das Ergebnis in Byte an.

Cache-Größe (in Byte): _____

Der Prozessor lädt nun sequentiell die Daten der folgenden Byteadressen (führende Nullen werden nicht mit angegeben):

```
t=1:  0xc25
t=2:  0x6db
t=3:  0x768
t=4:  0x6d0
t=5:  0xf80
t=6:  0x0a1
t=7:  0xaf3
t=8:  0xfc7
t=9:  0xa90
t=10: 0x111
```

Gehen Sie von folgendem Zustand des Cache zum Zeitpunkt $t = 0$ aus.

t=0	V	Tag
0	0	...00000
1	0	...11000
2	1	...00000
3	1	...11010
4	1	...11111
5	1	...01100
6	1	...01110
7	0	...10101

(c) Geben Sie den Zustand des Cache nach jedem Speicherzugriff in den folgenden Tabellen an und bestimmen Sie die Hitrate.
Hinweis: Es reicht aus, wenn Sie nur die Zeilen der Tabelle, die sich ändern, angeben.

Adresse=0x25c

t=1	V	Tag
0		
1		
2		
3		
4		
5		
6		
7		

Adresse=0x6db

t=2	V	Tag
0		
1		
2		
3		
4		
5		
6		
7		

Adresse=0x768

t=3	V	Tag
0		
1		
2		
3		
4		
5		
6		
7		

Adresse=0x6d0

t=4	V	Tag
0		
1		
2		
3		
4		
5		
6		
7		

Adresse=0xf8a

t=5	V	Tag
0		
1		
2		
3		
4		
5		
6		
7		

Adresse=0x0a1

t=6	V	Tag
0		
1		
2		
3		
4		
5		
6		
7		

Adresse=0xaf3

t=7	V	Tag
0		
1		
2		
3		
4		
5		
6		
7		

Adresse=0xfc7

t=8	V	Tag
0		
1		
2		
3		
4		
5		
6		
7		

Adresse=0xa90

t=9	V	Tag
0		
1		
2		
3		
4		
5		
6		
7		

Adresse=0x111

t=10	V	Tag
0		
1		
2		
3		
4		
5		
6		
7		

Hitrate: _____

Adresse=

t=	V	Tag
0		
1		
2		
3		
4		
5		
6		
7		

Adresse=

t=	V	Tag
0		
1		
2		
3		
4		
5		
6		
7		

Tabelle 1: Ersatztabelle, **ungültige Lösungen streichen**!

(d) Wie hoch ist die durchschnittliche Zugriffszeit auf den Speicher, wenn der Speicherzugriff auf den Cache 20ns und der Speicherzugriff auf den Hauptspeicher 200ns beträgt? Geben Sie auch den Rechenweg an.

Durchschnittliche Zugriffszeit: _____

Nehmen Sie an, der Cache sei leer und es sei folgender Ausschnitt des Hauptspeichers gegeben (führende Nullen werden nicht mit angegeben):

Adresse	Data
10 0000 1000	ff
10 0000 1001	9c
10 0000 1010	ad
10 0000 1011	c8
10 0000 1100	f0
10 0000 1101	da
10 0000 1110	aa
10 0000 1111	4c

Adresse	Data
10 0001 0000	01
10 0001 0001	d3
10 0001 0010	fa
10 0001 0011	c6
10 0001 0100	21
10 0001 0101	01
10 0001 0110	dd
10 0001 0111	df

Adresse	Data
10 0001 1000	1c
10 0001 1001	c3
10 0001 1010	d4
10 0001 1011	cd
10 0001 1100	25
10 0001 1101	e3
10 0001 1110	e2
10 0001 1111	7d

Adresse	Data
10 0010 0000	02
10 0010 0001	13
10 0010 0010	7c
10 0010 0011	51
10 0010 0100	00
10 0010 0101	a0
10 0010 0110	f4
10 0010 0111	22

(e) Es soll nun auf das Wort an der Adresse `0x217` zugegriffen werden. Geben Sie alle Daten an, die in den Cache geladen werden (nach jedem Byte getrennt durch ein Komma):

Übertragene Daten: _____

_____

_____

Konzeptpapier: Falls der Platz unter den einzelnen Aufgaben nicht ausreicht, können Sie diese Seiten für Zwischenrechnungen nutzen. Bitte Lösung und Lösungsweg eindeutig mit der Aufgabennummer markieren!