

Übungsklausur Rechnerarchitekturen

Klausurvariante 1

July 10, 2025

Prüfungsfach:	Rechnerarchitekturen
Semester:	Platzhalter-Semester
Prüfungsdauer:	90 Minuten
Maximale Punktzahl:	90 Punkte
Erlaubte Hilfsmittel:	Taschenrechner

Name:

Matrikelnummer:

Unterschrift:

Viel Erfolg!

Aufgabe 1: Leistung, Compiler und CPI

a)

- Computer A: 2GHz clock, 10s CPU time
- Entwurf von Computer B
 - Ziel: 6s CPU time
 - Hardware Ingenieur: kann die Taktrate erhöhen, aber dabei nehmen die Taktzyklen um Faktor 1,2 zu
- Wie schnell muss die Taktrate von B sein?

b)

Aufgabe 1.7

[15] <1.6> Compiler können einen erheblichen Einfluss auf die Performanz einer Applikation haben. Nehmen Sie an, dass Compiler A für ein Programm einen dynamischen Befehlszähler von $1,0E9$ ergibt und eine Ausführungszeit von 1,1 s hat, während Compiler B einen dynamischen Befehlszähler von $1,2E9$ ergibt und eine Ausführungszeit von 1,5 s hat.

- Bestimmen Sie unter der Annahme, dass der Prozessor eine Taktzeit von 1 ns hat, den mittleren CPI-Wert für jedes Programm.
- Angenommen, das kompilierte Programm läuft auf zwei verschiedenen Prozessoren. Wenn die Ausführungszeiten auf den beiden Prozessoren gleich sind, um wie viel schneller ist dann die Taktung des Prozessors, auf dem der Code von Compiler A läuft gegenüber der Taktung des Prozessors, auf dem der Code von Compiler B läuft?
- Es ist ein neuer Compiler entwickelt worden, der nur $6,0E8$ Befehle verwendet und einen mittleren CPI-Wert von 1,1 hat. Wie groß ist die Be-

schleunigung, die sich durch Verwendung dieses neuen Compilers anstelle von A oder B auf dem gleichen Prozessor erreichen lässt?

Aufgabe 2: MIPS

a)

Aufgabe 2.27

[5] <2.7> Übersetzen Sie den folgenden C-Code in MIPS-Assemblercode. Verwenden Sie dabei eine minimale Anzahl von Befehlen. Nehmen Sie an, dass die Werte von a, b, i und j in den Registern \$s0, \$s1, \$t0 bzw. \$t1 stehen. Nehmen Sie außerdem an, dass das Register \$s2 die Basisadresse des Feldes D enthält.

```
for(i=0; i<a; i++)
    for(j=0; j<b; j++)
        D[4*j] = i + j;
```

b)

Aufgabe 2.28

[5] <2.7> Wie viele MIPS-Befehle sind nötig, um den C-Code aus Aufgabe 2.27 umzusetzen? Die Variablen a und b seien mit 10 bzw. 1 initialisiert und alle Elemente von D haben den Anfangswert 0. Wie viele MIPS-Befehle werden insgesamt ausgeführt, um die Schleife zu vervollständigen?

c)

Aufgabe 2.41

[5] <2.6, 2.10> Der aktuelle Wert des Befehlszählers sei 0x000000600. Können Sie mithilfe eines einzelnen Sprungbefehls die Befehlszähleradresse aus Aufgabe 2.39 erhalten?

0010 0000 0000 0001 0100 1001 0010 0100_B

d)

Aufgabe 2.2

[5] <2.2> Wie lautet die zu dem folgenden MIPS-Assemblercode gehörende C-Anweisung?

```
add f, g, h
add f, i, f
```

e)

Aufgabe 2.13

Nehmen Sie an, dass `$s0` den Wert 128_D enthält.

2.13.1 [5] <2.4> Für welche(n) Wertebereich(e) von `$s1` resultieren die Anweisungen `add $t0,$s0,$s1` in einem Überlauf?

2.13.2 [5] <2.4> Für welche(n) Wertebereich(e) von `$s1` resultieren die Anweisungen `sub $t0,$s0,$s1` in einem Überlauf?

2.13.3 [5] <2.4> Für welche(n) Wertebereich(e) von `$s1` resultieren die Anweisungen `sub $t0,$s1,$s0` in einem Überlauf?

Aufgabe 3: Arithmetik

a)

Aufgabe 3.12

[20] <3.3> Verwenden Sie eine ähnliche Tabelle wie Tabelle 3.2, um das Produkt der vorzeichenlosen oktalen 6-Bit-Ganzzahlen 62 und 12 mit der in Abbildung 3.2 beschriebenen Hardware zu berechnen. Zeigen Sie für jeden Schritt die Registerinhalte.

b)

Aufgabe 3.10

[10] <3.2> Nehmen Sie an, dass 151 und 214 vorzeichenbehaftete dezimale 8-Bit-Ganzzahlen sind, die im Zweierkomplement-Format gespeichert werden. Berechnen Sie $151 - 214$ mittels Sättigungsarithmetik. Schreiben Sie das Ergebnis in Dezimaldarstellung.

c)

Aufgabe 3.4

[5] <3.2> Was ist $4365 - 3412$, wenn diese Werte vorzeichenlose 12-Bit-Oktalzahlen darstellen? Schreiben Sie das Ergebnis in Oktalдарstellung.

Aufgabe 4: Pipelining

a)

Aufgabe 4.15

Wie wichtig es ist, eine gute Sprungvorhersage zu haben, hängt davon ab, wie häufig bedingte Verzweigungen ausgeführt werden. Zusammen mit der Genauigkeit der Sprungvorhersage bestimmt dies, wie viel Zeit für Leerläufe aufgrund falscher Sprungvorhersagen aufgewendet wird. Nehmen Sie bei dieser Aufgabe an, die Unterteilung der dynamischen Befehle in verschiedene Befehlskategorien wie folgt aussieht:

R-Typ	beq	jmp	lw	sw
40 %	25 %	5 %	25 %	5 %

Gegeben sind außerdem die folgenden Genauigkeiten der Sprungvorhersage:

immer verzweigt	nie verzweigt	2-Bit
45 %	55 %	85 %

4.15.1 [10] <4.8> Leerlaufakte aufgrund falscher Sprungvorhersagen vergrößern den CPI-Wert. Wie groß ist der Betrag, der mit dem immer-verzweigt-Prädiktor zum CPI-Wert hinzu kommt? Nehmen Sie an, dass die Sprungergebnisse in der EX-Stufe bestimmt werden, dass es keine Datenkonflikte gibt und dass keine Warteplätze verwendet werden.

4.15.2 [10] <4.8> Wiederholen Sie Aufgabe 4.15.1 für den nie-verzweigt-Prädiktor.

4.15.3 [10] <4.8> Wiederholen Sie Aufgabe 4.15.1 für den 2-Bit-Prädiktor.

4.15.4 [10] <4.8> Wie groß ist die Beschleunigung, die wir mit dem 2-Bit-Prädiktor erreichen würden, wenn wir die Hälfte der Sprungbefehle so umformen könnten, dass ein Sprungbefehl durch einen ALU-Befehl ersetzt wird? Nehmen Sie an, dass korrekt und falsch vorhergesagte Befehle die gleiche Wahrscheinlichkeit haben, ersetzt zu werden.

4.15.5 [10] <4.8> Wie groß ist die Beschleunigung, die wir mit dem 2-Bit-Prädiktor erreichen würden, wenn wir die Hälfte der Sprungbefehle so umformen könnten, dass jeder Sprungbefehl durch zwei ALU-Befehle ersetzt wird? Nehmen Sie an, dass korrekt und falsch vorhergesagte Befehle die gleiche Wahrscheinlichkeit haben, ersetzt zu werden.

4.15.6 [10] <4.8> Manche Sprungbefehle sind viel besser vorhersagbar als andere. Wenn wir wissen, dass 80 % aller ausgeführten Sprungbefehle leicht vorhersage Schleifenrücksprünge sind, die immer korrekt vorhergesagt werden, wie gut ist dann die Genauigkeit des 2-Bit-Prädiktors bei den verbleibenden 20 % der Sprungbefehle?

Aufgabe 5: Caching

a) **5.2** 3, 180, 43, 2, 191, 88, 190, 14, 181, 44, 186, 253

Aufgabe 5.7	5.3										
		Adresse									
		0	4	16	132	160	1024	30	140	3100	180

In dieser Aufgabe betrachten wir den Einfluss unterschiedlicher Cache-Entwürfe, insbesondere durch Vergleich von assoziativen Caches mit direkt abgebildeten Caches aus Abschnitt 5.4. Dabei soll die Adressfolge aus Aufgabe 5.2 verwendet werden.

5.7.1 [10] <5.4> Zeigen Sie unter Verwendung der Zugriffe aus Übung 5.3 den endgültigen Cache-Inhalt für einen dreifach satzassoziativen Cache mit 2-Wort-Blöcken und einer Gesamtgröße von 24 Wörtern. Verwenden Sie eine LRU-Ersetzung. Geben Sie für jeden Zugriff die Index-Bits, die Tag-Bits und die Block-Offset-Bits an, und ebenso, ob es sich um einen Treffer oder um einen Fehlzugriff handelt.

5.7.2 [10] <5.4> Zeigen Sie unter Verwendung der Zugriffe aus Übung 5.3 den endgültigen Cache-Inhalt für einen vollständig assoziativen Cache mit 1-Wort-Blöcken und einer Gesamtgröße von acht Wörtern. Verwenden Sie eine LRU-Ersetzung. Geben Sie für jeden Zugriff die Index-Bits, die Tag-Bits und die Block-Offset-Bits an, und ebenso, ob es sich um einen Treffer oder um einen Fehlzugriff handelt.

5.7.3 [15] <5.4> Bestimmen Sie unter Verwendung der Zugriffe aus Aufgabe 5.2 die Fehlzugriffsrate für einen vollständig assoziativen Cache mit 2-Wort-Blöcken und einer Gesamtgröße von acht Wörtern. Verwenden Sie eine LRU-Ersetzung. Wie hoch ist die Fehlzugriffsrate unter Verwendung einer MRU-Ersetzung (Most Recently Used)? Wie hoch ist die bestmögliche Trefferrate für diesen Cache für eine beliebige Ersatzstrategie?

Multilevel-Caches stellen eine wichtige Technik dar, um den begrenzten Platz zu kompensieren, den ein primärer Cache bieten kann, während er gleichzeitig seine Geschwindigkeit beibehält. Betrachten Sie einen Prozessor mit den folgenden Parametern:

Basis-CPI, keine Speicherverzögerungen	1,5
Prozessorgeschwindigkeit	2 GHz
Hauptspeicherzugriffszeit	100 ns
Fehlzugriffsrate pro Befehl für den L1-Cache	7 %
Geschwindigkeit des direkt abgebildeten L2-Caches	12 Takte
allgemeine Fehlzugriffsrate bei dem direkt abgebildeten L2-Cache	3,5 %
Geschwindigkeit des 8-fach satzassoziativen L2-Caches	28 Takte
allgemeine Fehlzugriffsrate des 8-fach satzassoziativen L2-Caches	1,5 %

5.7.4 [10] <5.4> Berechnen Sie den CPI für den Prozessor mit den gegebenen Parametern. Verwenden Sie dazu 1) nur einen L1-Cache, 2) einen direkt abgebildeten L2-Cache und 3) einen achtfach satzassoziativen L2-Cache. Wie ändern sich diese Zahlen, wenn die Hauptspeicherzugriffszeit verdoppelt wird? Und wie, wenn sie halbiert wird?

5.7.5 [10] <5.4> Es ist möglich, eine noch tiefere Cache-Hierarchie als zwei Ebenen zu verwenden. Für den oben beschriebenen Prozessor mit direkt abgebildetem L2-Cache will ein Entwickler einen L3-Cache einführen, der 50 Takte für den Zugriff benötigt und die allgemeine Fehlzugriffsrate auf 1,3 % senkt. Führt dies zu einer besseren Leistung? Welche Vor- und Nachteile hat ein L3-Cache ganz allgemein?

5.7.6 [20] <5.4> In älteren Prozessoren wie etwa dem Intel Pentium oder dem Alpha 21264 war der L2-Cache extern angeordnet, also auf einem anderen Chip als der Hauptprozessor und der L1-Cache. Auf diese Weise konnten sehr große L2-Caches realisiert werden, aber die Zugriffslatenz war sehr viel höher, und die Bandbreite war in der Regel niedriger, weil der L2-Cache mit einer niedrigeren Frequenz betrieben wurde. Gehen Sie von einem 512 KiB großen L2-Cache auf einem separaten Chip aus, der eine allgemeine Fehlzugriffsrate von 4 % aufweist. Wenn jede zusätzlichen 512 KiB Cache die allgemeinen Fehlzugriffsraten um 0,7 % verschlechtern und der Cache eine Gesamtzugriffszeit von 50 Takten hatte, wie groß hätte dann der Cache sein müssen, um die Leistung des durch die Parameterliste spezifizierten direkt abgebildeten L2-Caches bzw. des achtfach satzassoziativen Caches zu erbringen?

b)

Aufgabe 5.6

In dieser Aufgabe betrachten wir die unterschiedlichen Arten, wie sich die Kapazität auf die Gesamtleistung auswirken kann. Im Allgemeinen ist die Cache-Zugriffszeit proportional zur Kapazität. Angenommen, Hauptspeichierzugriffe dauern 70 ns, und die Speichierzugriffe stellen 36 % aller Befehle. Die folgende Tabelle zeigt Daten für L1-Caches der beiden Prozessoren P1 und P2.

	L2-Größe	L2-Fehlzugriffsrate	L2-Trefferzeit
P1	2 KiB	8 %	0,66 ns
P2	4 KiB	6 %	0,90 ns

5.6.1 [5] <5.4> Angenommen, die L1-Trefferzeit bestimmt die Zyklendauern für P1 und P2. Wie hoch sind die jeweiligen Taktraten?

5.6.2 [5] <5.4> Wie hoch ist die AMAT für P1 und P2?

5.6.3 [5] <5.4> Gehen Sie von einem Basis-CPI von 1,0 aus. Wie hoch ist der Gesamt-CPI für P1 und P2? Welcher Prozessor ist schneller?

Für die nächsten drei Teilaufgaben betrachten wir das Hinzufügen eines L2-Caches zu P1, um die begrenzte Kapazität seines L1-Caches zu kompensieren. Verwenden Sie die L1-Cache-Kapazitäten und Trefferzeiten aus der obigen Tabelle. Die L2-Fehlzugriffsrate ist die lokale Fehlzugriffsrate.

L2-Größe	L2-Fehlzugriffsrate	L2-Trefferzeit
1 MiB	95 %	5,62 ns

5.6.4 [10] <5.4> Wie hoch ist die AMAT für P1, wenn ein L2-Cache eingeführt wird? Ist die AMAT mit L2-Cache besser oder schlechter?

5.6.5 [5] <5.4> Gehen Sie von einem Basis-CPI von 1,0 aus. Wie hoch ist der Gesamt-CPI für P1, nachdem der L2-Cache eingeführt wurde?

5.6.6 [10] <5.4> Welcher Prozessor ist schneller, nachdem P1 einen L2-Cache hat? Wenn P1 schneller ist, welche Fehlzugriffsrate bräuchte P2 dann für seinen L1-Cache, um der Leistung von P1 gleichzukommen? Wenn P2 schneller ist, welche Fehlzugriffsrate bräuchte P1 in seinem L1-Cache, um der Leistung von P2 gleichzukommen?