

Übungsklausur Rechnerarchitekturen

Klausurvariante 4

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)

Aufgabe 1.6

[20] <1.6> Betrachten wir zwei verschiedene Implementierungen der gleichen Befehlssatzarchitektur. Die Befehle können anhand ihres CPI-Werts in vier Klassen (A, B, C und D) unterteilt werden. P1 hat eine Taktfrequenz von 2,5 GHz und die CPI-Werte 1, 2, 3 und 3 für die vier Klassen. P2 hat eine Taktfrequenz von 3 GHz und die CPI-Werte 2, 2, 2 und 2.

Gegeben sei ein Programm mit einem dynamischen Befehlszähler von 1,6E6 Befehlen, die wie folgt auf die Klassen verteilt sind: 10 % in Klasse A, 20 % in Klasse B, 50 % in Klasse C und 20 % in Klasse D. Welcher Prozessor ist schneller, P1 oder P2?

- a. Was sind die globalen CPI-Werte der beiden Implementierungen?
- b. Bestimmen Sie für beide Fälle die erforderlichen Taktzyklen.

Aufgabe 2: MIPS

a)

Aufgabe 2.22

[5] <2.6> Schreiben Sie eine minimale Sequenz von MIPS-Assemblerbefehlen, die das gleiche bewirkt wie die folgende C-Anweisung. Es sei `$t1=A`, `$t2=B`, und `$s1` sei die Basisadresse von `C`.

```
A = C[0] << 4;
```

b)

Aufgabe 2.31

[5] <2.7> Implementieren Sie den folgenden C-Code in MIPS-Assembler. Wie viele MIPS-Befehle sind insgesamt nötig, um die Funktion auszuführen?

```
int fib(int n){
    if (n==0)
        return 0;
    else if (n==1)
        return 1;
    else
        return fib(n-1) + fib(n-2);
}
```

c)

Aufgabe 2.47

Angenommen, für ein gegebenes Programm sind 70 % der ausgeführten Befehle arithmetische Befehle, 10 % Lade-/Speicherbefehle und 20 % Sprungbefehle.

2.47.1 [5] <2.19> Bestimmen Sie den mittleren CPI für diesen Befehlsmix unter der Annahme, dass ein arithmetischer Befehl zwei Zyklen erfordert, ein Lade-/Speicherbefehl sechs Zyklen und ein Sprungbefehl drei Zyklen.

2.47.2 [5] <2.19> Wenn eine Leistungssteigerung von 25 % erreicht werden soll, wie viele Zyklen darf dann ein arithmetischer Befehl im Mittel beanspruchen, wenn Lade-/Speicherbefehle und Sprungbefehle überhaupt nicht verbessert werden?

2.47.3 [5] <2.19> Wenn eine Leistungssteigerung von 50 % erreicht werden soll, wie viele Zyklen darf dann ein arithmetischer Befehl im Mittel beanspruchen, wenn Lade-/Speicherbefehle und Sprungbefehle überhaupt nicht verbessert werden?

d)

Aufgabe 2.4

[5] <2.2, 2.3> Wie lautet die zu dem folgenden MIPS-Assemblercode gehörende C-Anweisung? Nehmen Sie an, dass die Variablen *f*, *g*, *h*, *i* und *j* den Registern *\$s0*, *\$s1*, *\$s2*, *\$s3* bzw. *\$s4* zugewiesen sind. Nehmen Sie außerdem an, dass sich die Basisadressen der Felder *A* und *B* in den Registern *\$s6* bzw. *\$s7* befinden.

```
sll    $t0, $s0, 2      # $t0 = f * 4
add    $t0, $s6, $t0    # $t0 = &A[f]
sll    $t1, $s1, 2      # $t1 = g * 4
add    $t1, $s7, $t1    # $t1 = &B[g]
lw     $s0, 0($t0)      # f = A[f]
addi   $t2, $t0, 4
lw     $t0, 0($t2)
add    $t0, $t0, $s0
sw     $t0, 0($t1)
```

Aufgabe 3: Arithmetik

a)

Aufgabe 3.23

[10] <3.5> Notieren Sie eine Binärdarstellung der Dezimalzahl 63,25, wenn das IEEE-754-Single-Precision-Format angenommen wird.

b)

Aufgabe 3.11

[10] <3.2> Nehmen Sie an, dass 151 und 214 vorzeichenlose 8-Bit-Ganzzahlen sind. Berechnen Sie $151 + 214$ mittels Sättigungsarithmetik. Schreiben Sie das Ergebnis in Dezimaldarstellung.

Aufgabe 4: Pipelining

a)

Instruction	Cycle										
	CC 1	CC 2	CC 3	CC 4	CC 5	CC 6	CC 7	CC 8	CC 9	CC10	CC11
sll R3,R1,2											
add R3,R2,R3											
lw R4,0(R3)											
lw R5,4(R3)											
sw R5,0(R3)											
sw R4,4(R3)											

1. Gebe Datenabhängigkeiten von diesem code an ohne forwarding
2. Gebe Datenabhängigkeiten von diesem code an mit vollem forwarding

Aufgabe 5: Caching

a)

Aufgabe 5.17

Die Cache-Kohärenz bezieht sich auf die Sicht, die mehrere Prozessoren auf einen bestimmten Cache-Block haben. Die folgende Tabelle zeigt zwei Prozessoren und ihre Lese/Schreiboperationen für zwei unterschiedliche Wörter eines Cache-Blocks X (anfänglich ist $X[0] = X[1] = 0$). Nehmen Sie an, dass die Größe der Ganzzahlen 32 Bit ist.

P1	P2
$X[0] ++$; $X[1] = 3$;	$X[0] = 5$; $X[1] += 2$;

5.17.1 [15] <5.10> Listen Sie die möglichen Werte dieses Cache-Blocks für eine korrekte Implementierung des Cache-Kohärenzprotokolls auf. Listen Sie mindestens einen weiteren Wert des Blocks auf, der möglich ist, wenn das Protokoll keine Cache-Kohärenz sicherstellt.

5.17.2 [15] <5.10> Listen Sie für ein Snooping-Protokoll eine gültige Operationsfolge für jeden Prozessor/Cache auf, um die obigen Lese-/Schreiboperationen auszuführen.

5.17.3 [10] <5.10> Wie viele Cache-Fehlzugriffe entstehen im besten und im ungünstigsten Fall, um die aufgezeigten Lese-/Schreibbefehle auszuführen?

Speicherkonsistenz bezieht sich auf die Ansichten mehrerer Datenelemente. Die folgende Tabelle zeigt zwei Prozessoren und ihre Lese-/Schreiboperationen für unterschiedliche Cache-Blöcke (A und B sind anfänglich 0).

P1	P2
$A = 1$; $B = 2$; $A += 2$; $B ++$;	$C = B$; $D = A$;

5.17.4 [15] <5.10> Listen Sie die möglichen Werte von C und D für eine Implementierung auf, die die Konsistenzannahmen von Seite 502 sicherstellt.

5.17.5 [15] <5.10> Listen Sie mindestens ein weiteres Wertepaar für C und D auf, wenn diese Annahmen nicht eingehalten werden.

5.17.6 [15] <5.3, 5.10> Mit welchen Kombinationen aus Schreibstrategien und Schreibreservierungsstrategien können Sie die Implementierung des Protokolls vereinfachen?