

Übungsklausur Rechnerarchitekturen

Klausurvariante 8

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 1.15

[5] <1.8> Wenn ein Programm so angepasst wird, dass es auf mehreren Prozessoren in einem Multiprozessorsystem läuft, dann beinhaltet die Ausführungszeit auf jedem Prozessor die Rechenzeit und die zusätzliche Zeit, die für gesperrte kritische Abschnitte und/oder den Austausch von Daten zwischen den einzelnen Prozessoren erforderlich ist.

Angenommen, ein Programm braucht eine Ausführungszeit von $t = 100$ s auf einem Prozessor. Wenn p Prozessoren eingesetzt werden, braucht jeder Prozessor die Zeit t/p und, unabhängig von der Anzahl der Prozessoren, zusätzlich 4 s für Kommunikation und Synchronisation. Berechnen Sie die pro

Prozessor benötigte Ausführungszeit für ein System mit 2, 4, 8, 16, 32, 64 und 128 Prozessoren. Listen Sie für jeden dieser Fälle die Beschleunigung sowie das Verhältnis zwischen tatsächlicher Beschleunigung und idealisierter Beschleunigung (Vernachlässigung des Overheads) gegenüber dem System mit nur einem Prozessor auf.

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 2.40

[5] <2.6, 2.10> Der aktuelle Wert des Befehlszählers sei 0x00000000. Können Sie mithilfe eines einzelnen Sprungbefehls die Befehlszähleradresse aus Aufgabe 2.39 erhalten?

c)

Aufgabe 2.12

Nehmen Sie an, dass die Register \$s0 und \$s1 die Werte 0x80000000 bzw. 0xD0000000 enthalten.

2.12.1 [5] <2.4> Was ist der Wert von \$t0 für den folgenden Assemblercode?

```
add $t0, $s0, $s1
```

2.12.2 [5] <2.4> Hat das Ergebnis in \$t0 den gewünschten Wert oder gab es einen Überlauf?

2.12.3 [5] <2.4> Wie lautet für die oben spezifizierten Register \$s0 und \$s1 der Wert \$t0 für den folgenden Assemblercode?

```
sub $t0, $s0, $s1
```

2.12.4 [5] <2.4> Hat das Ergebnis in \$t0 den gewünschten Wert oder gab es einen Überlauf?

2.12.5 [5] <2.4> Wie lautet für die oben spezifizierten Register \$s0 und \$s1 der Wert \$t0 für den folgenden Assemblercode?

```
add $t0, $s0, $s1
add $t0, $t0, $s0
```

2.12.6 [5] <2.4> Hat das Ergebnis in \$t0 den gewünschten Wert oder gab es einen Überlauf?

d)

Aufgabe 2.9

[5] <2.2, 2.3> Übersetzen Sie den folgenden C-Code in MIPS. Nehmen Sie an, dass die Variablen *f*, *g*, *h*, *i* und *j* den Registern *\$s0*, *\$s1*, *\$s2*, *\$s3* bzw. *\$s4* zugeordnet sind. Nehmen Sie an, dass die Basisadressen der Felder *A* und *B* in den Registern *\$s6* bzw. *\$s7* stehen. Nehmen Sie an, dass die Elemente der Felder *A* und *B* 4-Byte-Wörter sind.

$$B[8] = A[i] + A[j];$$

Aufgabe 3: Arithmetik

a)

Aufgabe 3.41

[10] <3.5> Verwenden Sie das IEEE 754 Gleitkommaformat, um ein Bitmuster aufzuschreiben, das $-1/4$ repräsentiert. Können Sie $-1/4$ exakt darstellen?

b)

Aufgabe 3.1

[5] <3.2> Was ist $5ED4 - 07A4$, wenn diese Werte vorzeichenlose 16-Bit-Hexadezimalzahlen darstellen? Schreiben Sie das Ergebnis in Hexadezimaldarstellung.

Aufgabe 4: Pipelining

a)

Aufgabe 4.8

In dieser Aufgabe untersuchen wir, wie das Pipelining die Taktzykluszeit des Prozessors beeinflusst. Bei den einzelnen Teilaufgaben wird vorausgesetzt, dass die individuellen Stufen des Datenpfades die folgenden Latenzen haben:

IF	ID	EX	MEM	WB
250 ps	350 ps	150 ps	300 ps	200 ps

Außerdem nehmen wir an, dass die vom Prozessor ausgeführten Befehle wie folgt verteilt sind:

alu	beq	lw	sw
45 %	20 %	20 %	15 %

4.8.1 [5] <4.5> Wie lang ist der Taktzyklus in einem Prozessor mit Pipelining und wie lang ist er in einem Prozessor ohne Pipelining?

4.8.2 [10] <4.5> Wie groß ist die Gesamtlatenz eines lw-Befehls in einem Prozessor mit Pipelining? Wie groß ist sie in einem Prozessor ohne Pipelining?

4.8.3 [10] <4.5> Wenn wir eine Stufe eines Datenpfades mit Pipelining in zwei neue Stufen aufteilen könnten, von denen jede die halbe Latenz der ursprünglichen Stufe hat, welche Stufe würden Sie dann für diese Aufteilung wählen und wie lang ist der neue Taktzyklus des Prozessors?

4.8.4 [10] <4.5> Angenommen, es gibt keine Leerläufe oder Konflikte. Wie ist dann die Auslastung des Datenspeichers?

4.8.5 [10] <4.5> Angenommen, es gibt keine Leerläufe oder Konflikte. Wie ist dann die Auslastung des Schreibregister-Ports der Registereinheit?

4.8.6 [30] <4.5> Anstatt einer Einzeltaktorganisation können wir auch eine Mehrtaktorganisation verwenden, bei der jeder Befehl mehrere Takte beansprucht, jedoch ein Befehl fertig ist, bevor ein anderer geholt wird. Bei dieser Organisation durchläuft ein Befehl nur Stufen, die er tatsächlich braucht (beispielsweise benötigt ST nur vier Stufen, weil kein Rückschreiben nötig ist). Vergleichen Sie die Taktzeiten und Ausführungszeiten mit Einzeltakt-, Mehrtakt- und Pipeline-Organisation.

Aufgabe 5: Caching

a)

Aufgabe 5.8

MTBF (Mean Time Between Failures), MTTR (Mean Time To Replacement) und MTTF (Mean Time To Failure) sind nützliche Maße für die Bewertung der Zuverlässigkeit und Verfügbarkeit einer Speicherressource. Diese Aufgabe beschäftigt sich mit diesen Maßen, wobei zwei Geräte betrachtet werden, von denen eines eine MTTF von 3 Jahren und das andere eine MTTR von einem Tag hat.

5.8.1 [5] <5.5> Berechnen Sie die MTBF für beide Geräte.

5.8.2 [5] <5.5> Berechnen Sie die Verfügbarkeit für beide Geräte.

5.8.3 [5] <5.5> Wie verhält sich die Verfügbarkeit, wenn die MTTR gegen 0 geht? Ist dies eine realistische Situation?

5.8.4 [5] <5.5> Wie verhält sich die Verfügbarkeit, wenn die MTTR sehr hoch wird, d. h. wenn ein Gerät schwer zu reparieren ist? Bedeutet dies, dass das Gerät eine geringe Verfügbarkeit hat?