

Übungsklausur Rechnerarchitekturen

Klausurvariante 6

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 1.13

Ein anderer Fallstrick, der in Abschnitt 1.10 genannt ist, besteht in der Erwartung, dass sich die Gesamtleistung eines Computers verbessert, wenn nur ein einzelner Leistungsparameter verbessert wird. Betrachten wir einen Computer, auf dem ein Programm läuft, welches insgesamt 250 s benötigt, wobei 70 s für das Ausführen von Gleitkommaoperationen aufgewendet werden, 85 s für Lade-/Speicherbefehle und 40 s für Sprungbefehle.

1.13.1 [5] <1.10> Um wie viel wird die Gesamtzeit reduziert, wenn der Zeitaufwand für Gleitkommaoperationen um 20 % reduziert wird?

1.13.2 [5] <1.10> Um wie viel wird der Zeitaufwand für Ganzzahloperationen reduziert, wenn die Gesamtzeit um 20 % reduziert wird?

1.13.3 [5] <1.10> Kann die Gesamtzeit um 20 % reduziert werden, indem man lediglich die Zeit für Sprungbefehle reduziert?

Aufgabe 2: MIPS

a)

Aufgabe 2.24

[5] <2.7> Angenommen, der Programmzähler ist auf `0x2000 0000` gesetzt. Ist es möglich, den Sprungassemblerbefehl von MIPS (j) zu verwenden, um den Programmzähler auf die Adresse `0x4000 0000` zu setzen? Ist es möglich, den Befehlszähler mit dem MIPS-Assemblerbefehl `branch-on-equal (beq)` auf die gleiche Adresse zu setzen?

b)

Aufgabe 2.38

[5] <2.9> Gegeben sei der folgende Code:

```
lbu $t0, 0($t1)
sw $t0, 0($t2)
```

Nehmen Sie an, dass das Register `$t1` die Adresse `0x1000 0000` enthält und das Register `$t1` die Adresse `0x1000 0010`. Beachten Sie, dass die MIPS-Architektur eine Big-Endian-Adressierung benutzt. Nehmen Sie an, dass der an der Adresse `0x1000 0000` gespeicherte Wert `0x11223344` ist. Welcher Wert wird an der Adresse gespeichert, auf die das Register `$t2` zeigt?

c)

- C code:
`g = h + A[8];`
 - `g` in `$s1`, `h` in `$s2`, Basisadresse von `A` in `$s3`

d)

Aufgabe 2.7

[5] <2.3> Zeigen Sie, wie der Wert `0xabcdef12` im Speicher einer Little-Endian- und einer Big-Endian-Maschine angeordnet wäre. Nehmen Sie an, dass die Daten beginnend mit der Adresse 0 gespeichert sind.

Aufgabe 3: Arithmetik

a)

Aufgabe 3.24

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

b)

Aufgabe 3.6

[5] <3.2> Nehmen Sie an, dass 185 und 122 vorzeichenlose dezimale 8-Bit-Ganzzahlen sind. Berechnen Sie $185 - 122$. Gibt es dabei einen Überlauf, einen Unterlauf oder nichts von beidem?

Aufgabe 4: Pipelining

a)

Aufgabe 4.3

Wenn Prozessorentwickler eine mögliche Verbesserung des Datenpfads untersuchen, dann hängt die Entscheidung gewöhnlich von dem Verhältnis zwischen Kosten und Performanzsteigerung ab. In den drei Teilaufgaben dieser Aufgabe wird angenommen, dass wir mit einem Datenpfad wie dem in Abbildung 4.2 dargestellten beginnen, wo I-Mem, Add, Mux, ALU, Regs, D-Mem und Steuerblöcke die Latenzen 400 ps, 100 ps, 30 ps, 120 ps, 200 ps, 350 ps bzw. 100 ps haben. Die Kosten betragen 1000, 30, 10, 100, 200, 2000 bzw. 500.

Nun nehmen wir an, dass ein Multiplizierer zur ALU hinzugefügt wird. Durch dieses Hinzufügen kommen 300 ps zur Latenz und 600 zu den Kosten der ALU dazu. Das Ergebnis ist, dass 5 % weniger Befehle ausgeführt werden, da der MUL-Befehl nicht mehr emuliert werden muss.

4.3.1 [10] <4.1> Wie lang ist ein Taktzyklus mit und ohne die Verbesserung?

4.3.2 [10] <4.1> Um wie viel schneller wird der Prozessor durch die Verbesserung?

4.3.3 [10] <4.1> Vergleichen Sie das Verhältnis der Kosten zur Performanz für den Fall mit und ohne die Verbesserung.

Aufgabe 5: Caching

a)

Aufgabe 5.2

Caches sind wichtig, um eine leistungsstarke Speicherhierarchie für Prozessoren aufzubauen. Nachfolgend finden Sie eine Liste von 32-Bit-Speicheradressen, die als Wortadressen angegeben sind.

3, 180, 43, 2, 191, 88, 190, 14, 181, 44, 186, 253

5.2.1 [10] <5.3> Geben Sie für jeden dieser Zugriffe die Binäradresse, das Tag und den Index bei einem direkt abgebildeten Cache mit 16 Ein-Wort-Blöcken an. Geben Sie außerdem an, ob die einzelnen Zugriffe Treffer oder Fehlzugriffe sind, vorausgesetzt, der Cache ist anfangs leer.

5.2.2 [10] <5.3> Geben Sie für jede dieser Zugriffe die Binäradresse, das Tag und den Index bei einem direkt abgebildeten Cache mit Zwei-Wort-Blöcken und einer Gesamtgröße von acht Blöcken an. Geben Sie außerdem an, ob die einzelnen Zugriffe Treffer oder Fehlzugriffe sind, vorausgesetzt, der Cache ist anfangs leer.

5.2.3 [20] <5.3, 5.4> Sie sollen den Cache-Entwurf für die vorgegebenen Zugriffe optimieren. Es sind drei direkt abgebildete Cache-Entwürfe möglich, alle mit insgesamt acht Datenwörtern: C1 verwendet Ein-Wort-Blöcke, C2 verwendet Zwei-Wort-Blöcke und C3 verwendet Vier-Wort-Blöcke. Welcher Cache-Entwurf ist im Hinblick auf die Fehlzugriffsrate am besten? Welcher Cache-Entwurf ist am besten, wenn die Fehlzugriffsverzögerung 25 Takte beträgt und C1 eine Zugriffszeit von 2 Takten, C2 von 3 Takten und C3 von 5 Takten hat?

Es gibt viele verschiedene Entwurfsparameter, die für die Gesamtleistung eines Caches von Bedeutung sind. Die nachfolgende Tabelle listet Parameter für die verschiedenen direkt abgebildeten Cache-Entwürfe auf.

Cache-Datengröße: 32 KiB

Cache-Blockgröße: 2 Wörter

Cache-Zugriffszeit: 1 Taktzyklus

5.2.4 [15] <5.3> Berechnen Sie die Gesamtanzahl der Bits, die für die in der Tabelle aufgelisteten Caches erforderlich sind, wobei 32-Bit-Adressen vorausgesetzt sind. Bestimmen Sie anhand dieser Gesamtgröße die Gesamtgröße des nächstliegenden direkt abgebildeten Caches mit 16-Wort-Blöcken gleicher Größe oder größer. Erklären Sie, warum der zweite Cache trotz seiner größeren Datengröße möglicherweise eine langsamere Leistung bietet als der erste.

5.2.5 [20] <5.3, 5.4> Erzeugen Sie eine Folge von Leseanforderungen, die eine niedrigere Fehlzugriffsrate auf einem zweifach satzassoziativen 2 KiB-Cache haben als der in der Tabelle angegebene Cache. Geben Sie eine mögliche Lösung an, durch die der in der Tabelle angegebene Cache eine gleiche oder niedrigere Fehlzugriffsrate als der 2 KiB-Cache bekommen würde. Diskutieren Sie Vor- und Nachteile einer solchen Lösung.

5.2.6 [15] <5.3> Die auf Seite 412 gezeigte Formel zeigt die typische Methode, einen direkt abgebildeten Cache zu indizieren, insbesondere (Blockadresse) modulo (Anzahl der Blöcke im Cache). Gehen Sie von 32-Bit-Adressen und 1024 Blöcken im Cache aus. Betrachten Sie eine andere Indizierungsfunktion, nämlich (Blockadresse[31:27] XOR Blockadresse[26:22]). Ist es möglich, sie zur Indizierung eines direkt abgebildeten Caches zu verwenden? Erklären Sie gegebenenfalls, warum das so ist, und diskutieren Sie Änderungen, die eventuell an dem Cache vorgenommen werden müssen. Erklären Sie, warum das gegebenenfalls nicht möglich ist.