

Grundlagen der Informatik III

Wintersemester 2010/2011

Wolfgang Heenes, Patrik Schmittat



TECHNISCHE
UNIVERSITÄT
DARMSTADT

5. Aufgabenblatt

29.11.2010

Hinweis: Der Schnelltest und die Aufgaben sollen in den Übungsgruppen bearbeitet werden. Die Hausaufgaben sind in der Kalenderwoche 49 (06.12. bis 10.12.) bei den Tutoren in **physikalischer Form** (handschriftlich oder gedruckt) abzugeben. Bei allen Abgaben ist der Name des Tutors und die Übungsgruppe deutlich anzugeben. Bei Teamabgaben wird nur eine Lösung eingereicht, die alle Namen der Teammitglieder enthält.

Schicken Sie Ihre Lösungen von Programmieraufgaben *zusätzlich* zur schriftlichen Abgabe per E-Mail an Ihren Tutor. Kommentieren Sie Ihren Quellcode.

Aufgabe 1: Schnelltest

Fragen	Antworten
1. Welche Variable ist ein optimaler Indikator für die Performanz eines beliebigen Programms?	<input type="checkbox"/> Anzahl der Taktzyklen für Programmausführung. <input type="checkbox"/> Anzahl der Instruktionen des Programms. <input type="checkbox"/> Durchschnittliche Anzahl der Taktzyklen pro Instruktion. <input type="checkbox"/> Durchschnittliche Anzahl der Taktzyklen pro Sekunde.
2. Welche Aussagen zu Bewertungskriterien für Prozessoren sind richtig?	<input type="checkbox"/> Die CPU-Ausführungszeit ergibt sich aus den verwendeten Operationen und der Taktung der CPU. <input type="checkbox"/> Die CPU-Ausführungszeit ist die gemessene Zeit, welche die CPU real zum Ausführen der Codezeilen eines Programms benötigt. <input type="checkbox"/> Die CPU-Systemzeit ist die Zeit, welche die CPU für Betriebssystemaufgaben verwendet. <input type="checkbox"/> Die Benutzerantwortzeit ist ein geeignetes Maß für CPU-Performanz. <input type="checkbox"/> Rechner mit hohem Durchsatz haben auch kürzere Ausführungszeiten als Rechner mit niedrigem Durchsatz.
3. Welche Elemente einer Programmausführung fließen nicht in die <i>user-CPU-time</i> mit ein?	<input type="checkbox"/> Festplattenzugriffe <input type="checkbox"/> Verzögerung durch Cache-Misses <input type="checkbox"/> I/O-Verzögerung durch Hardware <input type="checkbox"/> CPU-Ausführungszeit
4. Welche Elemente einer Programmausführung fließen in die <i>Benutzerantwortzeit</i> mit ein?	<input type="checkbox"/> I/O-Verzögerung durch Hardware <input type="checkbox"/> Verzögerung durch Cache-Misses <input type="checkbox"/> CPU-Ausführungszeit <input type="checkbox"/> Festplattenzugriffe

Aufgabe 2: Gleitkommazahlen

Gegeben sei ein Rechner, der mit 8-Bit Gleitkommazahlen arbeitet. Diese 8 Bit teilen sich auf in 1 Bit für das Vorzeichen, 3 Bit für den Exponenten und 4 Bit für die Mantisse:

$$\boxed{V} \boxed{E} \boxed{E} \boxed{E} \boxed{M} \boxed{M} \boxed{M} \boxed{M} \quad r = -1^V \cdot (1 + M) \cdot 2^{E-\text{Bias}}$$

- Wie groß ist der Bias, der für die Darstellung des Exponenten verwendet wird?
- Wandeln Sie die Zahl 6,25 in die beschriebene Darstellung um.
- Wandeln Sie die Zahl $-0,78125$ in die beschriebene Darstellung um.
- Multiplizieren Sie die beiden Repräsentation (Darstellung der 8 Bit Gleitkommazahl) von 6,25 und $-0,78125$. Geben Sie das Ergebnis in Gleitkommadarstellung an und wandeln Sie es in die Dezimalzahldarstellung um. Hinweise zur Multiplikation von Gleitkommazahlen sind durch die Suche in Büchern, Datenbanken (ULB) und dem Internet zu finden.

Aufgabe 3: Gleitkommazahl nach IEEE 754

Gegeben ist eine Gleitpunktzahl G im IEEE 754 Standard (einfache Genauigkeit) in Hexadezimaldarstellung.

$$G = C080\ 0000_{16}$$

Welchen dezimalen Wert repräsentiert die die Darstellung von G ?

Aufgabe 4: Leistungsbewertung

Folgende Größen sind im Zusammenhang mit der Leistungsbewertung gegeben:

T	CPU-Zeit in Sekunden
f	Taktfrequenz in Zyklen pro Sekunde
t	Taktzykluszeit in Sekunden pro Zyklus
MIPS	Millionen von Instruktionen pro Sekunde
CPI	mittlere Anzahl Zyklen pro Instruktion
B	Anzahl Instruktionen
Z	CPU-Zeit in Zyklen

- Leiten Sie die Formeln der gesuchten Größen aus den angegebenen Größen her.
 - gesucht: T, gegeben: Z, t
 - gesucht: CPI, gegeben: Z, MIPS, T
 - gesucht: T, gegeben: f, CPI, B
 - gesucht: B, gegeben: T, MIPS
 - gesucht: Z, gegeben: T, CPI, MIPS
- Zwei unterschiedliche Prozessoren mit jeweils 2.4 GHz und 2.233 GHz Taktfrequenz bearbeiten eine Berechnung, die insgesamt aus $9 \cdot 10^7$ Instruktionen besteht. Der erste Rechner benötigt 14 Zyklen pro Instruktion, der zweite Rechner 12 Zyklen pro Instruktion. Welcher Rechner ist schneller fertig?
- Die Operationen einer komplexen Berechnung benötigen zusammengenommen $7.7 \cdot 10^7$ Rechenzyklen. Wie lange dauert die Ausführung auf einem Prozessor mit 3.2 GHz Taktfrequenz?

- d) Ein Rechner, der eine Leistung von 2.3 MIPS hat, benötigt für eine komplexe Berechnung 1350 ms. Wie viele Instruktionen werden dabei verarbeitet?
- e) Obiger Rechner hat eine Taktfrequenz von 750 MHz. Wie hoch ist die mittlere Anzahl an Rechenzyklen pro Instruktion?

Aufgabe 5: Performanzvergleich

Betrachtet werden zwei Hardware-Implementierungen M1 und M2 eines Befehlssatzes mit vier Befehlsklassen Add, Jmp, Mul und Shft. Die Taktfrequenz für M1 ist 1.133 GHz und für M2 750 MHz. Desweiteren sind die Daten aus den folgenden Tabelle gegeben.

Befehlsklasse	CPI für M1	CPI für M2
Add	2	1
Jmp	2	2
Mul	5	3
Shft	1	1

Befehlsklasse	Häufigkeit
Add	45%
Jmp	15%
Mul	25%
Shft	15%

- a) Ein Programm P1 weist die in der Tabelle angegebene Verteilung der Befehlshäufigkeiten auf. Wie groß ist die CPI-Rate von M1 und M2 bei der Ausführung von P1?
- b) Wie ist die relative Performanz zwischen M1 und M2 bei Ausführung von P1?
- c) Das Programm P2 enthält jeweils gleich viele Befehle aus den vier Befehlsklassen. Wie hoch ist die mittlere Anzahl der Zyklen pro Befehl (CPI) bei Ausführung von P2 auf M1 und M2?
- d) Wie ist die relative Performanz zwischen M1 und M2 bei Ausführung von P2?

Hausaufgabe 1: Nachbildung einer FPU mit Integerarithmetik (6 Punkte)

In dieser Aufgaben sollen Sie die bisher nativ genutzte FPU (Floating Point Unit) softwaretechnisch mittels IA32-Assemblercode emulieren. Das bedeutet, dass Sie die Grundrechenarten +, * für IEEE 754 single precision in IA32-Assemblercode implementieren sollen. Die Spezialfälle (vgl. Vorlesung 11, Folie 42) müssen nicht berücksichtigt zu werden.

Es steht Ihnen für diese Aufgabe ein Rahmencode zur Verfügung. **Nutzen Sie diesen Rahmencode.** Er enthält eine Ausgabe für die Gleitkommazahlen und alle Unterprogramme, die Sie schreiben sollen. Natürlich dürfen für diese Aufgabe keine FPU-Befehle genutzt werden. Weiterhin sollten Sie darauf achten die vorgegebenen Konventionen der Unterprogrammtechnik einzuhalten. Kommentieren Sie ihren Code ausführlich!

- a) Vervollständigen Sie zuerst die Unterprogramme `sign`, `exponent`, `significand`, um die `fltprint`-Methode testen zu können. Die Unterprogramme sollen folgende Funktionen übernehmen:
- `sign`: Vorzeichen der Gleitkommazahl extrahieren
 - `exponent`: Exponent der Gleitkommazahl extrahieren
 - `significand`: Mantisse der Gleitkommazahl extrahieren

Danach sollten Sie in der Lage sein die `fltprint`-Methode zu nutzen und ein float auf der Konsole auszugeben.

- b) Implementieren Sie nun die Unterprogramme für Addition und Multiplikation. Nutzen Sie an dieser Stelle falls notwendig die zur Verfügung stehenden Unterprogramme `sign`, `exponent`, `significand`.

Hausaufgabe 2: Arithmetischer Ausdruck mit Gleitkommazahlen (4 Punkte)

Gegeben ist der folgende arithmetische Ausdruck:

$$((a - x) + b) / ((c - d) * (e * f - (g / h + i)))$$

- a) Stellen Sie diesen Ausdruck als binären Baum dar. Ergänzen Sie zunächst im Ausdruck Klammern, die einzelne Terme zusammenfassen. Beachten Sie dabei die Berechnungsfolge. **Hinweis:** Operatoren gleicher Priorität sollen von links nach rechts ausgewertet werden.
- b) Geben Sie die LR-Postorder-Reihenfolge der Knoten an.
- c) Schreiben Sie ein IA32-Assemblerprogramm (ATT- oder Intel-Syntax) welches die LR-Postorder-Darstellung in Maschinenbefehle abbildet. Dabei sollen 32-Bit Gleitkommazahlen und die x87 FPU verwendet werden. Halten Sie Zwischenergebnisse ausschließlich auf dem FPU-Stack. Fehlersituationen wie Überlauf und Division durch Null sollen **nicht** berücksichtigt werden. Das Ergebnis soll am Ende in **st(0)** stehen und ausgegeben werden. Es dürfen keine General Purpose Register (%eax, usw.) verwendet werden (außer für die Ausgabe). Die Variablen dürfen nicht verändert werden und es dürfen, außer den Eingabewerte a, b, c, d, e, f, g, h, i, x, keine zusätzlichen Variablen im **.DATA**-Bereich vereinbart werden. Testen Sie Ihr Programm mit folgenden Werten.

$$a = 2.3, b = 0.7, c = 4.7, d = 1.7, e = 3.0, f = 2.0, g = 9.0, h = 3.0, i = 1.0, x = 1.0$$

Hinweis: Eine ausführliche Beschreibung der x87 FPU finden Sie unter <https://ics.ra.informatik.tu-darmstadt.de/svn/Material/Referenz/Intel/253665.pdf>, Abschnitt 8-1.