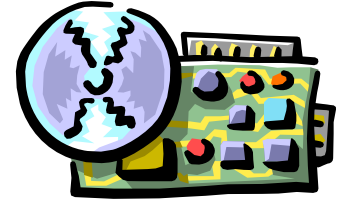




# TI II: Rechnerarchitektur SS 2007 Übungsblatt Nr. 4



Prof. Dr.-Ing. Jochen Schiller, AG Technische Informatik, Freie Universität Berlin

**Ausgabe am 01.06.2007 — Abgabe spätestens 22.06.2006, 12:00 Uhr**

Bitte bei der Abgabe beide Namen/Matr.Nr. der Mitglieder einer Gruppe, NUMMER DER ÜBUNG/TEILAUFGABE und DATUM auf den Lösungsblättern **nicht vergessen!** Darauf achten, dass die Lösungen beim richtigen Tutor/der richtigen Tutorin abgegeben werden.

**Zu spät abgegebene Lösungen werden nicht mehr berücksichtigt!**

## 1. Aufgabe: Sprungvorhersage (3 Punkte)

Betrachten wir 3 Vorhersagemethoden:

- Sprünge werden nie ausgeführt
- Sprünge werden immer ausgeführt
- Dynamische Vorhersage mit einer Vorhersagegenauigkeit im Durchschnitt von 90%

Nehmen wir an, bei diesen Methoden entstehen keine Kosten, wenn die Vorhersage stimmt, und es entstehen Kosten in Höhe von 2 Zyklen, wenn die Vorhersage nicht stimmt.

Welches Verfahren eignet sich für die folgenden Sprünge am besten?

- Ein Sprung, der mit einer Häufigkeit von 5% ausgeführt wird.
- Ein Sprung, der mit einer Häufigkeit von 95% ausgeführt wird.
- Ein Sprung, der mit einer Häufigkeit von 70% ausgeführt wird.

	nie ausgeführt	immer ausgeführt	dynamische Vorhersage
5%			
70%			
95%			

## 2. Aufgabe: Assembler (6 Punkte)

- Entspricht das *goto* in der Sprache C der Assembler Sprunganweisung *jmp, j, etc*? Falls nicht: Worin besteht der Unterschied und wodurch ist er bedingt? (2 Punkte)
- Wie stellt sich ein Funktionsaufruf (z.B. `var0 = myFunction(var1, var2);`) auf Instruktionsebene dar. (4 Punkte)

## 3. Aufgabe: Fakultät (5 Punkte)

Programmieren Sie (MIPS Assembler) die Fakultätsfunktion ( $fak(n) = 1 * 2 * \dots * (n-1) * n$ ) rekursiv und iterativ und vergleichen Sie diese Implementierungen hinsichtlich des Speicherbedarfs und der Anzahl ausgeführter Befehle. Die Eingabe ( $n$ ) steht in  $\$t0$  und das Ergebnis soll in  $\$t1$  stehen.

## 4. Aufgabe: Sprungvorhersage - Assembler (6 Punkte)

Gehen Sie von einer 5-stufigen DLX Pipeline ohne Forwarding, Sprungvorhersage, Flushing-Möglichkeit, etc. aus. Die Pipeline berechnet nur was sie zum bearbeiten bekommt und unterstützt uns insofern bei Problemen in keiner Weise. Optimieren Sie folgendes Programmfragment bezüglich der Control-Hazards und fügen Sie falls notwendig NOPs ein:

original	Optimiert
<pre>add \$t0, \$t0, \$t1 sub \$t1, \$t1, \$t2 bne \$t3, \$t4, label1 slti \$t4, \$t5, 2 label1: add \$t5, \$t6, \$t2 beq \$t0, \$zero, label2 srl \$t0, \$t0, \$t1 label2: add \$t8, \$t8, \$t2 add \$t3, \$t3, \$t2 mul \$t4, \$t4, \$t2 add \$t5, \$t5, \$t2 beq \$t7, \$t8, label3 add \$t0, \$t0, \$zero label3:</pre>	

## 5. Aufgabe: Gauß-Summe (10 Punkte)

Schreiben Sie drei (MIPS-) Assembler-Routinen zur Berechnung der Gauß-Summe einer Zahl:

- 1) `gs_iter`, soll die Gauß-Summe iterativ berechnen (2 Punkte)
- 2) `gs_rec`, soll die Gauß-Summe rekursiv berechnen (2 Punkte)
- 3) `gs_ohne`, soll ohne Iteration oder Rekursion auskommen (2 Punkte)

Vergleichen Sie für Ihre Lösungen jeweils den Aufwand. (4 Punkte)

Geben Sie dazu die Verhältnisse der benötigten Assembler-Befehle der drei Lösungen in Abhängigkeit des Eingabeparameters  $n$ , der initial in Register `$t0` steht, an. Das Ergebnis soll in Register `$t1` stehen.

(Gauß-Summe:  $gs(n) = 1+2+\dots+n$ ,  $n>0$ ; Summe der natürlichen Zahlen von 1 bis  $n$ ).

## 6. Aufgabe: Pipelining (7 Punkte)

- Was bedeutet Latenz bei einer Pipeline? (2 Punkte)
- Wie wird der Durchsatz einer Pipeline definiert? (2 Punkte)
- Warum braucht ein Lade-/Speicherbefehl eine EX-Phase (Execution-Phase)? (1 Punkt)
- Was ist der Unterschied zwischen einer Sprungvorhersage mit 2-Bit Sättigungszähler und einer Sprungvorhersage mit 2-Bit Hysterese Schema? (2 Punkte)

## 7. Aufgabe: Unterbrechungen (6 Punkte)

- Was sind Unterbrechungen und warum werden diese benötigt?
- Wann können diese zum Beispiel auftreten? Sind Unterbrechungen häufig auf einem normalen Rechner?
- Was muss vor und nach der Unterbrechungsbehandlung gemacht werden?
- Woher weiß ein Prozessor, welche Unterbrechungsbehandlungsroutine ausgeführt werden soll?
- Kann eine Unterbrechungsbehandlung wiederum unterbrochen werden? Warum und zu welchen Problemen kann dies führen?
- Wann bietet sich Polling als Alternative an?