



Prof. Dr. Rolf Drechsler, drechsler@informatik.uni-bremen.de, MZH 3510  
Dr. Nicole Drechsler nd@informatik.uni-bremen.de, MZH 3550

3. Übungsblatt zur Vorlesung

# Technische Informatik 1

## Aufgabe 1

(4 Punkte)

Optimiere das folgende Programmstück unter der Annahme, dass

- $R1 > 0$  ist und
- lediglich der Wert aus Register  $R4$  in zeitlich nachfolgenden Programmbereichen verwendet wird.

```
(S0) LI R2, a           ; R2 := a (konstant)
(S1) BGEZ R1, S3      ; springe nach S3, falls R1 ≥ 0
(S2) LI R2, b           ; R2 := b (konstant)
(S3) SUB R4, R3, R2     ; R4 := R3 – R2
(S4) ADD R3, R2, R4     ; R3 := R2 + R4
(S5) LI R5, c           ; R5 := c (konstant)
(S6) MUL R4, R4, R5     ; R4 := R4 * R5
```

Begründe Deine Überlegungen.

## Aufgabe 2

(6 Punkte)

Unter der Annahme, dass in einem sequentiellen Programm die Anweisung  $S_i$  vor der Anweisung  $S_j$ ,  $0 \leq i < j \leq 6$ , steht, kann man *Datenabhängigkeiten* folgendermaßen klassifizieren:

- *True Dependence* (read after write):  $S_j$  liest eine Variable, die in  $S_i$  beschrieben wird.
  - *Anti Dependence* (write after read):  $S_j$  schreibt auf eine Variable, die in  $S_i$  gelesen wird.
  - *Output Dependence* (write after write):  $S_j$  schreibt auf eine Variable, die auch in  $S_i$  beschrieben wird.
- a) Betrachte das Programmstück aus Aufgabe 1. Bestimme alle Datenabhängigkeiten in diesem Programmstück. Unterscheide dabei zwischen True, Anti und Output Dependencies. (Ohne einschränkende Annahmen bezüglich der Ausführung zu machen.)
- b) Wie kann die Information über Datenabhängigkeiten genutzt werden?

## Aufgabe 3

(6 Punkte)

Betrachte noch einmal das Programmstück aus Aufgabe 1 (Ohne einschränkende Annahmen bezüglich der Ausführung). Die einzelnen Befehle werden in einer fünfstufigen Befehlspipeline (Befehl holen, Befehl dekodieren, Operanden holen, Operation ausführen, Operand speichern) verarbeitet. Erst am Ende der Operand-speichern-Phase ist ein Schreibvorgang in das entsprechende Zielregister abgeschlossen. Bei Sprungbefehlen wird der Befehlszähler während der Operation-ausführen-Phase auf den neuen Wert gesetzt.

- a) Wieviele und welche Pipeline-Konflikte können auftreten?
- b) Behebe alle Pipeline-Konflikte durch Einfügen einer minimalen Anzahl von NOP-Befehlen.

#### **Aufgabe 4**

(4 Punkte)

- a) Zur Lösung von Control-Hazards wird Branch-Prediction verwendet. In welchen Fällen greift dieses Verfahren zum Vermeiden von Pipeline-Konflikten besonders gut? Geben Sie ein Beispiel an! Was ist der Nachteil bei der Verwendung von Branch-Prediction?
- b) Welche softwareseitigen Lösungen existieren für Data Hazards? Wo werden diese realisiert? Gibt es auch hardwaremäßige Lösungen?

**Abgabetermin: zu Beginn der Vorlesung am 15.05.2008**