



Prof. Dr. Rolf Drechsler, drechsler@informatik.uni-bremen.de, MZH 3510
Dr. Nicole Drechsler nd@informatik.uni-bremen.de, MZH 3485

3. Übungsblatt zur Vorlesung

Technische Informatik 1

Aufgabe 1

(4 Punkte)

Vereinfache die Ausdrücke des folgenden Programmstücks unter der Annahme, dass

- lediglich der Wert aus Register $R1$ in zeitlich auf S_5 folgenden Instruktionen verwendet wird und
- für die zwei Konstanten a und b immer $a > b$ gilt.

(S_0) LI $R1, a$; $R1 := a$ (konstant)
 (S_1) LI $R3, b$; $R3 := b$ (konstant)
 (S_2) SUB $R4, R3, R1$; $R4 := R3 - R1$
 (S_3) BGEZ $R4, S_0$; springe nach S_0 , falls $R4 \geq 0$
 (S_4) MUL $R4, R2, R3$; $R4 := R2 * R3$
 (S_5) SUB $R1, R4, R1$; $R1 := R4 - R1$

Begründe Deine Überlegungen.

Aufgabe 2

(6 Punkte)

Unter der Annahme, dass in einem sequentiellen Programm die Anweisung S_i vor der Anweisung S_j , $0 \leq i < j \leq 5$, 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? Gib 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 07.05.2009