Prof. Dr. Rolf Drechsler, drechsler@informatik.uni-bremen.de, MZH 3510
Dipl.-Inform. Daniel Große, grosse@informatik.uni-bremen.de, MZH 3490

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.

(S_0)	LI $R2$, a	; $R2 := a$ (konstant)
(S_1)	BGEZ $R1$, S_3	; springe nach S_3, falls $R1 \geq 0$
(S_2)	LI $R2$, b	; $R2 := b$ (konstant)
(S_3)	SUB $R4$, $R3$, $R2$	; $R4 := R3 - R2$
(S_4)	ADD $R3$, $R2$, $R4$	; $R3 := R2 + R4$
(S_5)	LI $R5$, c	; $R5 := c$ (konstant)
(S_6)	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_1 vor der Anweisung S_2 steht, kann man *Datenabhängigkeiten* folgendermaßen klassifizieren:

- *True Dependence* (read after write): S_2 liest eine Variable, die in S_1 beschrieben wird.
- *Anti Dependence* (write after read): S_2 schreibt auf eine Variable, die in S_1 gelesen wird.
- *Output Dependence* (write after write): S_2 schreibt auf eine Variable, die auch in S_1 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) Wo werden die Informationen über Datenabhängigkeiten benötigt?

Aufgabe 3 (6 Punkte)

Betrachte noch einmal das Programmstück aus Aufgabe 1 ohne einschränkende Annahmen. 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? 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 09.05.2005