



3. Übungsblatt zur Vorlesung

Technische Informatik I

Aufgabe 1

a) Zeige per Induktion über m : Die Laufzeit $L_P(m)$ eines Programms P mit m Instruktionen auf einem Rechner mit einer k -stufigen Pipeline beträgt

$$L_P(m) = t + (m - 1) \cdot t/k.$$

b) Begründe die folgende Formel für den Speedup s eines Programms P bei einer k -stufigen Pipeline:

$$s = \frac{m \cdot t}{t + (m - 1) \cdot t/k}$$

Annahmen: (1) Die Ausführung einer Instruktion benötigt die Zeit t . (2) Bei der Ausführung von P treten keine Hazards auf.

Aufgabe 2

Gegeben sei ein Programm P bestehend aus m Instruktionen. Es ist dabei $m = 50$ für die Demoversion, $m = 1000$ für die Sharewareversion und $m = 250000$ für die Vollversion. P soll auf einem Rechner mit einer 7-stufigen Pipeline ausgeführt werden, bei der jede Stufe gleich schnell ist. Glücklicherweise konnte nachgewiesen werden, daß keinerlei Abhängigkeiten (=Hazards) in P auftreten. Wie groß ist jeweils der Speedup gegenüber der Ausführung ohne Pipeline? Gegen welchen Wert konvergiert der Speedup, wenn die Programmlänge m gegen unendlich geht?

Aufgabe 3

In einer Weiterentwicklung P' des Programms P aus Aufgabe 2 fliegen während der Applikation im Hintergrund zufallsgesteuert Moorföhner umher. Kaum hat das Entwicklerteam voller Stolz dieses neue Feature präsentiert, werden Untersuchungen veröffentlicht, aus denen hervorgeht, daß bei P' aufgrund von falschen Verzweigungsvorhersagen (*branch prediction*) in b Fällen die Pipeline geleert werden muß, wobei b folgendermaßen von der Programmlänge m abhängt:

$$b(m) = \begin{cases} \frac{1}{20}m & : 1 \leq m < 2000 \\ \frac{1}{10}m - 120 & : m \geq 2000 \end{cases}$$

und für das Leeren der Pipeline genau zwei Takte benötigt werden. Wie groß ist der Speedup jetzt noch für die Demo-, Standard- und Vollversion, wenn man davon ausgeht, daß

- a) die Programmlänge m durch das Entfernen von unnützen Codezeilen bei der Weiterentwicklung beibehalten werden konnte und
- b) das Programm weiterhin auf einem Rechner mit 7-stufiger Pipeline (wie in Aufgabe 2) ausgeführt wird?

Aufgabe 4

Betrachten wir erneut das folgende sequentielle Programmstück:

```
S1: JMP S3 ;springe nach S3
S2: LOAD a,R4 ;R4:=a]
S3: LOAD b,R1 ;R1:=b]
S4: LOAD c,R2 ;R2:=c]
S5: MUL R1,R2,R1 ;R1:= R1.R2
S6: ADD R1,R2,R2 ;R2:= R1+R2
```

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 treten auf?
- b) Beheben sie diese Konflikte durch geeignetes Einfügen von NOP-Befehlen.

Aufgabe 5

Beim Entwurf eines Rechners ist man sich unklar darüber, ob man ihn mit einem *direct-mapped Cache* (DM), einem *2-fach-assoziativen Cache* (2A) oder einem *vollassoziativen Cache* (VA) ausstatten soll.

Die geschätzten technischen Daten für die zur Verfügung stehenden Caches:

Cache	Trefferrate	Zugriffszeit des Cache	Zugriffszeit zum Hauptspeicher
DM	97.1%	23.9ns	196ns
2A	96.2%	21.6ns	200ns
VA	96.9%	22.1ns	198ns

Für welchen Cache sollte man sich entscheiden, wenn das Hauptkriterium die durchschnittliche Zugriffszeit des Cache beim Lesen ist.

Hinweis: Es müssen 3 von 5 Aufgaben bearbeitet werden.
Abgabetermin: Dienstag, 14.11.2000, 14:00Uhr