Rechnerorganisation - 2. Klausurvorbereitung

Auer Thomas, Stefan Haan

25. Jänner 2018

Inhaltsverzeichnis

5 Übungsblatt 5

5.1 Multi-Cycle Datenpfad: lw-Instruktion

Erläutern Sie die Ausführung der lw-Instruktion (load word) für den Multi-Cycle-Datenpfad. Welche Vorteile bietet der Multi-Cycle-Datenpfad gegenüber dem Single-Cycle-Datenpfad? Diskutieren Sie dies anhand dieses Befehls.

5.2 Grundlagen Pipelining

Gegeben seien vier unterschiedliche Prozessoren, die sich in der Anzahl der Pipelinestufen und der Taktrate unterscheiden:

Prozessor	Pipelinestufen	Taktrate
A	1	100 MHz
B	4	800 MHz
C	12	1,5 GHz
D	20	3,2 GHz

(a) Bestimmen Sie für jeden Prozessor die Latenz der einzelnen Instruktionen.

Wie lange dauert die Ausführung von 400.000 voneinander unabhängigen Instruktionen auf jedem der angeführten Prozessoren? Bestimmen Sie die Performance und den Speedup verglichen mit Prozessor A ohne Pipelining. (Sie können annehmen, dass es keine Stalls gibt.)

5.3 Pipelining: Graphische Darstellung

Beantworten Sie folgende Fragen anhand der Beispiel-Pipeline-Architektur der VO (Kapitel 3.2).

(a) Bestimmen Sie die Anzahl der Pipelinestufen, die Taktdauer und die Taktfrequenz der Beispiel-Pipeline unter Annahme der Angaben auf VO-Folie 3-44 (Ausführungszeiten der Funktionseinheiten). Wie lange dauert die Ausführung eines einzigen Befehls auf der Beispiel-Pipeline?

(b) Angenommen es treten keine Leertakte (stalls) auf, welchen Speedup erreicht die Beispiel-Pipeline aus a) gegenüber einem Single-Cycle Datenpfad, der aus den gleichen Funktionsregistern besteht?
Seite 1 von 2

(c) Auf der Pipeline wird folgende Befehlssequenz ausgeführt:

```
and     $10, $2, $3
sw      $11, 4($3)
```

Stellen Sie die Ausführung der oben angeführten Befehlssequenz durch die Beispiel-Pipeline wie auf VO-Folie 3-45 grafisch dar (untere Abbildung). Achten Sie insbesondere auf die zeitliche Anordnung der Zugriffe auf die Registereinheit! Wie lange dauert die Ausführung der Befehlssequenz?

5.4 Pipelining: Daten- und Kontrollabhängigkeiten

Gegeben sei folgendes Code-Fragment:

6 Übungsblatt 6

6.5 Loop Unrolling

Folgendes Codefragment wird auf einen Prozessor mit „Delayed Branching“ (1 Takt Branch Delay) ausgeführt. Die Latenzen zwischen abhängigen Befehlen sind in Tabelle 1 aufgelistet.

```
loop:  l.d    $f4   0($t0)
       sub.d  $f6   $f4    $f0
       l.d    $f8   0($t1)
       mul.d  $f10  $f6    $f8
       add.d  $f12  $f10   $f2
       s.d    $f12  0($t2)
       addi   $t2   $t2    -8
       addi   $t1   $t1    -8
       addi   $t0   $t0    -8
       bne    $t0   $t4    loop
       nop
```

Erzeugender Befehl	Benutzender Befehl	Zwischentakte
FP ALU operation	FP ALU operation	3
FP ALU operation	Store FP double	2
Load FP double	FP ALU operation	1
Load FP double	Store FP double	0
Load integer	Integer operation	1
Load integer	Branch	2
Integer operation	Integer operation	0
Integer operation	Branch	1

Tabelle 1: Latenzen zwischen abhängigen Befehlen

(a) **Identifizieren Sie alle Daten- und Kontrollabhängigkeiten, die Stalls verursachen. Wie viele Takte werden für ein Ergebniselement[1] benötigt?**

```
loop:      l.d    $f4   0($t0)
+1 Stall   sub.d  $f6   $f4    $f0
           l.d    $f8   0($t1)
+2 Stalls  mul.d  $f10  $f6    $f8
+3 Stalls  add.d  $f12  $f10   $f2
+2 Stalls  s.d    $f12  0($t2)
           addi   $t2   $t2    -8
           addi   $t1   $t1    -8
           addi   $t0   $t0    -8
+1 Stall   bne    $t0   $t4    loop
           nop
```

	Stalls	Instruktionen	Σ
Ergebniselement	9	11	20

[1] durch s.d gespeicherter Wert

(b) **Optimieren Sie den Code durch Umordnen von Befehlen so, dass er auf dem gegebenen Prozessor möglichst schnell ausgeführt wird. Wie viele Takte werden für die Verarbeitung eines Ergebniselements benötigt?**

```
loop:       l.d     $f4   0($t0)
            l.d     $f8   0($t1)
            sub.d   $f6   $f4     $f0
            addi    $t2   $t2     -8
            addi    $t1   $t1     -8
+1 Stall    mul.d   $f10  $f6     $f8
            addi    $t0   $t0     -8
+2 Stalls   add.d   $f12  $f10    $f2
+1 Stalls   bne     $t0   $t4     loop
            s.d     $f12  8($t2)
```

	Stalls	Instruktionen	Σ
Ergebniselement	4	10	14

(c) **Rollen Sie die Schleife zweimal ab und ordnen Sie den Code so um, dass er auf dem gegebenen Prozessor möglichst schnell ausgeführt wird. Wie viele Takte werden nun pro Ergebniselement benötigt?**

```
loop:  l.d    $f4   0($t0)
       l.d    $f14  -8($t0)
       sub.d  $f6   $f4     $f0
       sub.d  $f16  $f14    $f0
       l.d    $f8   0($t1)
       l.d    $f18  -8($t1)
       mul.d  $f10  $f6     $f8
       mul.d  $f20  $f16    $f8
       addi   $t2   $t2     -16
       addi   $t1   $t1     -16
       add.d  $f12  $f10    $f2
       add.d  $f22  $f20    $f22
       addi   $t0   $t0     -16
       s.d    $f12  16($t2)
       bne    $t0   $t4     loop
       s.d    $f22  8($t2)
```

	Stalls	Instruktionen	Σ
2 Elemente	0	16	16
1 Element	0	8	8

7 Übungsblatt 7

8 Übungsblatt 8

8.4 Speicherhierarchien 1

Tabelle 2 gibt die **Verzögerung** im Falle eines **erfolgreichen** Zugriffs auf die jeweilige Speicherebene an.

Der durchschnittliche CPI-Wert ohne Berücksichtigung der Speicherzugriffe betrage 1,3 (idealer CPI). **20%** aller Instruktionen greifen auf den Speicher zu.

Speicherebene	Hitrate	Verzögerung/Takte
L1	85%	4
L2	75%	12
RAM	100%	236

Tabelle 2: Speicherebenen und Verzögerung in Takten

a) **Durchschnittlicher CPI-Wert unter Berücksichtigung der Speicherzugriffe**

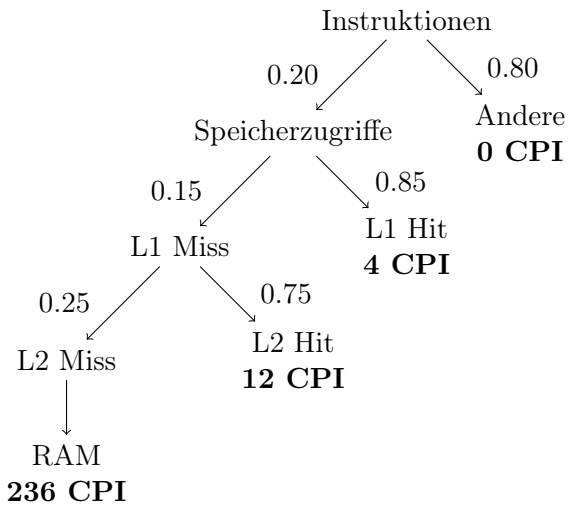

Abbildung 1: Wahrscheinlichkeiten der Speicherzugriffe mit ihren **Verzögerungen**

Berechne die durchschnittliche Verzögerung aus Abbildung 1 durch gewichtete Aufsummierung der Knoten.

$$\overline{\text{Verzögerung}} = 0.8 \cdot 0 + 0.2 \cdot (0.85 \cdot 4 + 0.15 \cdot (0.75 \cdot 12 + 0.25 \cdot 236)) = 2.72 \tag{1}$$

Addiere die durchschnittliche Verzögerung zum idealen CPI-Wert um den durchschnittlichen CPI-Wert unter Berücksichtigung der Speicherzugriffe zu erhalten.

$$\overline{\text{CPI}_{\text{gesamt}}} = \overline{\text{Verzögerung}} + 1.3 = 4.02 \tag{2}$$

b) **Prozent der Ausführungszeit, die der Prozessor auf Speicherzugriffe warten muss**

Die durch Speicherzugriffe verursachte Verzögerung ist $\overline{\text{Verzögerung}} = 2.72$. Berechne das Verhältnis zum durchschnittlichen CPI-Wert.

$$\frac{\overline{\text{Verzögerung}}}{\overline{\text{CPI}_{\text{gesamt}}}} = \frac{2.72}{4.02} \approx 0.677 \tag{3}$$

c) **Hitrate des L1-Caches, um einen durchschnittlichen CPI-Wert von 3 zu erhalten**

Die maximale Verzögerung, die nun durch Speicherzugriffe auftreten darf, ist $\overline{\text{Verzögerung}} = 3 - 1.3 = 1.7$. Man schreibe die Gleichung 1 um und löse nach der Hitrate des L1 Caches p_{L1}.

$$\overline{\text{Verzögerung}} = 0.8 \cdot 0 + 0.2 \cdot (p_{\text{L1}} \cdot 4 + (1 - p_{\text{L1}}) \cdot (0.75 \cdot 12 + 0.25 \cdot 236)) = 1.7 \tag{4}$$

$$p_{\text{L1}} \cdot 4 + (1 - p_{\text{L1}}) \cdot 68 = \frac{1.7}{0.2} \tag{5}$$

$$4 \cdot p_{\text{L1}} - 68 \cdot p_{\text{L1}} = 8.5 - 68 \tag{6}$$

$$p_{\text{L1}} = \frac{-59.5}{-64} \tag{7}$$

$$\approx 0.93 \tag{8}$$

9 Übungsblatt 9

10 Übungsblatt 10