



Technische Informatik II im SS 2005

Aufgaben zu den Tutorien in der Woche
vom 06. bis 08. Juni 2005

Dr.-Ing. T. Asfour

Haid-und-Neu-Str.7, Geb. 07.21
D-76131 Karlsruhe
Telefon: +49-721-608-7379
Fax: +49-721-608-8270
Email: asfour@ira.uka.de
<http://i61www.ira.uka.de/users/asfour/T1>

Lernziele:

- Einführung in die Grundlagen der Pipelineverarbeitung (Definitionen, Speed-up und Durchsatz)
- Welche Hardware-Komponenten sind notwendig zur (noch sequentiellen) Ausführung der in der Vorlesung behandelten 3 Typen der MIPS-Befehle?
- Datenpfade für die Befehlsausführung erklären.
- Daten- und Kontrollabhängigkeiten
- Software- und Hardware-Behebungsmaßnahmen

In den Folien ist die Vorgehensweise klar: Man nehme z. B. die Befehle vom R-Typ und legt die zu deren Ausführungsnotwendigen Komponenten und Datenpfade fest. Die Studenten sollten spätestens nach diesem Tutorium in der Lage sein, die Architektur problemlos abzuleiten bzw. fehlende Verbindungen zu ergänzen.

Aufgabe 1

Die folgende Programmsequenz soll in einer fünfstufigen DLX-Pipeline abgearbeitet werden:

```
S1:  MUL    R2, R2, R5  ;    R5 = R2 * R2
S2:  MUL    R4, R2, R1  ;    R1 = R4 * R2
S3:  ADD    R5, R3, R5  ;    R5 = R5 * R3
S4:  ADD    R1, R5, R1  ;    R1 = R1 * R5
```

1. Bestimmen Sie alle Datenabhängigkeiten in der gegebenen Programmsequenz.
2. Welche der gefundenen Abhängigkeiten führen in der obigen Programmsequenz zu Pipelinekonflikten?
3. Beseitigen Sie alle Pipelinekonflikte durch Einfügen von möglichst wenigen NOP-Befehlen.
4. Lösen Sie die Aufgabenteile 2 und 3, wenn die Bereitstellung von Operanden in der Stufe 2 und das Speichern von Ergebnissen in der Stufe 5 erst am Ende des Taktes abgeschlossen ist.

Aufgabe 2

Betrachten Sie das folgende MIPS-Programmstück:

```
# In Register $v0 steht die Adresse 0x10008000
lw  $t7, 0($v0)
lw  $s0, 4($v0)
add $s0, $s0, $t7      # $s0 = $s0 + $t7
mul $t7, $t7, $s0      # $t7 = $t7 * $s0
```

1. Bestimmen Sie alle Daten- und Kontrollabhängigkeiten in der gegebenen Programmstück.
2. Wieviele Pipelinekonflikte treten auf? Begründen Sie Ihre Antwort.
3. Unter der Voraussetzung, daß die auftretenden Pipelinekonflikte von der Hardware nicht erkannt werden, müssen die Pipelinekonflikte vom Compiler behandelt werden. Ergänzen Sie obiges Programmstück, so daß die auftretenden Pipelinekonflikte berücksichtigt werden.
4. Welche der von Ihnen getroffenen Maßnahmen im letzten Aufgabenteil sind noch erforderlich, falls die auftretenden Pipelinekonflikte von der Hardware erkannt und durch Load-Forwarding und Result-Forwarding behandelt werden.