

Rechnerorganisation im WS 2017/18

## 6. Übungsblatt

Abgabetermin: 18. Dezember, 13:15 Uhr

Prof. Dr. Wolfgang Karl  
Haid-und-Neu-Str. 7

Dr.-Ing. Ömer Terlemez  
Adenauerring 2, Geb. 50.20

Email: [ti@ira.uka.de](mailto:ti@ira.uka.de)  
Web: <http://ti.ira.uka.de>

### Aufgabe 1

(11 Punkte)

Das folgende Programmstück soll in der DLX-Pipeline abgearbeitet werden.

```
S1:          add  $t1, $zero, $zero
S2:          lw   $t3, 0x1500($zero)
S3:    loop:  lw   $t4, 0x5000($t1)
S4:          add  $t5, $t4, $t3
S5:          sw   $t5, 0x400($t1)
S6:          addi $t1, $t1, 4
S7:          subi $t2, $t1, 0x400
S8:          bnez $t2, loop
S9:    end:   srlr $t1, $t1, 2
S10:         sw   $t1, 0x2000($zero)
```

1. Bestimmen Sie alle echten Datenabhängigkeiten und alle Steuerflussabhängigkeiten im Programmstück. 4 P.
2. Gehen Sie davon aus, dass *Forwarding*-Techniken implementiert sind. Die weitere Methode zur Behebung von Pipelinekonflikten sei das Einfügen von NOP-Befehlen (*No Operation*) in den Befehlsstrom. 3 P.  
Ergänzen Sie das Programmstück durch das Einfügen von möglichst wenigen NOP-Befehlen, so dass alle Pipelinekonflikte behoben werden.
3. Was sind Struktur- oder Ressourcenkonflikte? Können diese bei der Ausführung des Programmstücks in der DLX-Pipeline auftreten? Begründen Sie Ihre Antwort. 2 P.
4. Warum können Ausgabeabhängigkeiten (*output dependence*) und Gegenabhängigkeiten (*anti-dependence*) in der DLX-Pipeline nicht zu Konflikten führen? 1 P.
5. Alle heutigen Prozessoren behandeln Steuerflusskonflikte durch die Hardware. Eine einfache Methode besteht in der Spekulation, dass bedingte Sprünge nicht genommen werden. 1 P.

Warum ist diese Methode ineffizient?

Aufgabe 2

(13 Punkte)

Gegeben sei folgendes Programmstück:

```
S1: addi $s0, $zero, 100
S2: lui  $t0, 0x2014
S3: ori  $t0, $t0, 0x1234
S4: lw   $s1, 4($t0)
S5: add  $s1, $s1, $s0
S6: srl  $s2, $s1, 2
S7: sw   $s2, 4($t0)
```

1. Bestimmen Sie alle echten Datenabhängigkeiten ( $\delta^t$ ) im Programmstück. 3 P.  
Geben Sie zu jeder Datenabhängigkeit die beiden beteiligten Befehle und das ursächliche Register an.
2. Das Programmstück wird nun auf einer DLX-Pipeline ausgeführt, die weder Forwarding noch andere Hardware-Mechanismen zur Verhinderung von Datenkonflikten realisiert. 3 P.  
Geben Sie nach jedem Taktschritt den Zustand der Pipeline (welcher Befehl ist in welcher Pipeline-Stufe?) und den Inhalt der Register  $\$s0$  und  $\$s1$  und  $\$t0$  an.  
Nehmen Sie an, dass die genannten Register mit 0 initialisiert sind und Schreibvorgänge in den Registersatz im jeweiligen Taktzyklus bereits abgeschlossen werden.
3. Das Programm soll auf einer DLX-Pipeline ausgeführt werden, für die keine Forwarding-Techniken implementiert sind. Fügen Sie dazu eine minimale Anzahl von Leerbefehlen (NOP-Instruktionen) ein, sodass keine der Datenabhängigkeiten zu Konflikten führt. 2 P.  
Verändern Sie die Reihenfolge der Befehle nicht.
4. Betrachten Sie nun eine DLX-Pipeline auf der Result Forwarding und Load Forwarding implementiert sind. Fügen Sie eine minimale Anzahl von Leerbefehlen (NOP-Instruktionen) ein, sodass keine der Datenabhängigkeiten zu Konflikten führt. 2 P.  
Verändern Sie die Reihenfolge der Befehle nicht.
5. Geben Sie die Anzahl der zur Ausführung notwendigen Taktzyklen für die sequentielle Ausführung und die (korrekte) Ausführung auf einer DLX-Pipeline ohne und mit Forwarding an. 3 P.  
Nehmen Sie für die sequentielle Ausführung an, dass die Ausführung jedes Befehls fünf Taktzyklen benötigt.  
Für die Ausführung auf einer DLX-Pipeline analysieren Sie die modifizierten Programmstücke, die Sie in den vorhergehenden Aufgabenteilen erstellt haben.

Aufgabe 3

(3 Punkte)

Gegeben seien die folgenden drei MIPS-Instruktionen:

1. `sub $s3, $s4, $s5`
2. `ori $s4, $zero, 42`
3. `lb $s4, 0x12345678`

Nennen Sie die drei Typen von Datenabhängigkeiten, die Ihnen bekannt sind.

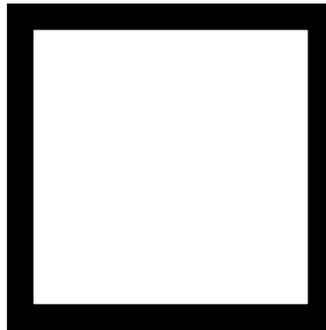
Geben Sie für jeden dieser drei Typen eine Sequenz von zwei der drei genannten Instruktionen an, die diese Abhängigkeit enthält.

Geben Sie an, welches Register jeweils von der Datenabhängigkeit betroffen ist.

Vorlesung Rechnerorganisation Wintersemester 2017/18

- Übungsblatt 6 -

Tutoriumsnummer



Name, Vorname: \_\_\_\_\_

Matrikelnummer: \_\_\_\_\_

Studiengang: \_\_\_\_\_

Name des Tutors: \_\_\_\_\_

**/27 Punkte**