



Exercise 9

16. Juni 2015

Abgabe: 23. Juni 2015, 10.00 Uhr

Problem 9.1: Pipeline

15 Points

Simulieren Sie den Ablauf des folgenden Codes, wenn er durch eine DLX-Pipeline läuft! Erweitern Sie dazu untenstehende Diagramm, welches angibt, welche Stufe jeder Befehl in welchem Takt durchläuft!

Die DLX-Pipeline besteht (wie die in der Vorlesung besprochene MIPS-Pipeline) aus den 5 Stufen *Instruction Fetch* (IF), *Instruction Decode* (ID), *Execute* (EX), *Memory Access* (MA) und *Write Back* (WB). In der EX-Phase berechnete Werte werden erst am Ende der WB-Phase in die Register zurückgeschrieben. Wenn ein darauffolgender Befehl in seiner EX-Phase diesen Wert für die Berechnung benötigt, muss die Pipeline daher so lange verzögert (*stalled*) werden, bis der entsprechende Wert in das Register zurückgeschrieben wurde. Auch `lw` und `sw` können erst mit der Ausführung beginnen, wenn alle Quellregister zur Verfügung stehen. Wie im Diagramm zu sehen ist, wird Register `r2` in Takt 6 (WB des zweiten Befehls) zurückgeschrieben. Da der `add`-Befehl dieses Register benötigt, muss die Pipeline so lange verzögert werden, bis `r2` in Takt 7 für die Ausführung (EX) zur Verfügung steht. Im Gegensatz zur Vorlesung wird hier angenommen, dass ein geschriebener Registerwert erst im nächsten Takt zum Lesen zur Verfügung steht.

```

1 lw  r1 , 0x1000(r0)
2 lw  r2 , 0x1004(r0)
3 add r3 , r1 , r2
4 sub r4 , r1 , r2
5 sw  0x100c(r0) , r4
6 sw  0x1008(r0) , r3
7 addi r5 , r0 , 0x10
8 sub  r6 , r0 , r5
9 sw  0x2000(r1) , r5
10 sw  0x2000(r2) , r6

```

1. Erweitern Sie das Diagramm um die benötigte Verzögerung!
2. Wieviele Takte lang muss die Pipeline verzögert werden?
3. Welche Beschleunigung wurde mit der Pipeline für diesen Code erreicht? Gehen Sie davon aus, dass für einen Takt mit Pipelining 200 ps benötigt werden. Ohne Pipelining dauert ein Takt 800 ps.

Instr.	Takt										
	1	2	3	4	5	6	7	8	9	10	11 ...
lw r1, 0x1000(r0)	IF	ID	EX	MA	WB						
lw r2, 0x1004(r0)		IF	ID	EX	MA	WB					
add r3, r1, r2			IF	ID			EX	MA	WB		
sub r4, r1, r2				IF			ID	EX	MA	WB	
...											

Problem 9.2: Pipeline Reorder

15 Points

Beschleunigen Sie das folgende Programm, so dass es eine DLX-Pipeline (vgl. vorherige Aufgabe) schneller durchläuft! Ordnen Sie dafür die einzelnen Anweisungen um, so dass es zu weniger Konflikten kommt, welche die Pipeline anhalten! Die Funktionalität des Programms muss natürlich dieselbe bleiben.

```

1 lw r1 , 0x1000 ( r0 )
2 lw r2 , 0x1004 ( r0 )
3 add r3 , r1 , r2
4 sub r4 , r1 , r2
5 sw 0x100c ( r0 ) , r4
6 sw 0x1008 ( r0 ) , r3
7 addi r5 , r0 , 0x10
8 sub r6 , r0 , r5
9 sw 0x2000 ( r1 ) , r5
10 sw 0x2000 ( r2 ) , r6

```

1. Geben Sie den umgeordneten Quelltext sowie ein Ablaufdiagramm an, aus dem hervorgeht, welcher Befehl in welcher Stufe und in welchem Takt durchläuft.
2. Wieviele Takte lang muss die Pipeline verzögert werden?
3. Welche Beschleunigung wurde mit der Pipeline für diesen Code erreicht? Gehen Sie davon aus, dass für einen Takt mit Pipelining 200 ps benötigt werden. Ohne Pipelining dauert ein Takt 800 ps.

**Die Fachschaft möchte Sie zum diesjährigen
 Sommerfest der Informatik
 am Freitag, 26.06.2015 ab 18 Uhr
 auf der WSI-Sonnenterasse (vor dem Raum A104)
 herzlich einladen!**

Total: 30 Points