

Rechnerorganisation im WS 2020/21

Musterlösungen zum 6. Übungsblatt

Prof. Dr. Jörg Henkel
Dr.-Ing. Lars Bauer
Roman Lehmann, M. Sc.
Haid-und-Neu-Str. 7,
Geb. 07.21 (Technologiefabrik)
Email: roman.lehmann@kit.edu

Lösung 1

(4 Punkte)

1. Dauer eines Befehls: $(50 + 60 + 60 + 60 + 50) \text{ ns} = 280 \text{ ns}$
 \Rightarrow Benötigte Zeit zur Ausführung von 1000 Befehlen: $280\,000 \text{ ns}$

1 P.

Taktperiode der Pipeline:

$$\tau = \max\{\tau_1, \tau_2, \dots, \tau_5\} + \tau_{reg} = (60 + 5) \text{ ns} = 65 \text{ ns}$$

1 P.

2. Dauer eines Befehls: $(65 \times 5) \text{ ns} = 325 \text{ ns}$
3. Benötigte Zeit zur Ausführung von 1000 Befehlen:
Anzahl Takte: $n + (k - 1) = 1000 + 5 - 1 = 1004$
Benötigte Zeit: $1004 \times 65 = 65\,260$
Alternativ:

1 P.

$$(65 \times 5 \times 1) + (65 \times 1 \times 999) = 65260 \text{ ns}$$

1 P.

4. Beschleunigung (*Speedup*):

$$\begin{aligned} \text{Beschleunigung} &= \frac{\text{sequentielle Ausführungszeit}}{\text{Ausführungszeit mit Pipeline}} \\ &= \frac{280000}{65260} = 4,29 \end{aligned}$$

Lösung 2

(6 Punkte)

1.
 - IF-Stufe (*instruction fetch*): 5 P.
Der Befehl wird aus dem Speicher geladen. Gleichzeitig wird der Wert des Befehlszählers um 4 erhöht und in das Befehlszählerregister zurückgeschrieben.
 - ID-Stufe (*instruction decode*):
Das Befehlswort wird dekodiert. Das Steuerwerk erzeugt die Steuersignale, die die nachfolgenden Komponenten beeinflussen. Das Register, welches die zu lesende Speicheradresse enthält, wird gelesen und der Wert weitergeleitet.
 - EX-Stufe (*execute*):
Die ALU berechnet die effektive Adresse für den Speicherzugriff, indem der Registerinhalt mit dem im Befehlswort angegeben Offset addiert wird.
 - MEM-Stufe (*memory*):
Der lesende Speicherzugriff wird durchgeführt.
 - WB-Stufe (*write-back*):
Das gelesene Datenwort wird in den Registersatz geschrieben.
2. In der ID-Stufe wird zusätzlich der zu speichernde Wert aus dem Registersatz ausgelesen. 1 P.

In der MEM-Stufe wird statt einem lesenden ein schreibender Speicherzugriff durchgeführt.

Die WB-Stufe wird untätig durchlaufen, da am Registersatz keine Veränderung vorgenommen werden muss.

Lösung 3

(15 Punkte)

1. Adresse der Speicherstelle: $0x20141238$ ($0x20141234 + 0x00000004$) 1 P.
2. Wert der Speicherstelle nach sequentieller Ausführung: 1 P.
 $\lfloor \frac{2014+100}{4} \rfloor = 528$
3. Echte Datenabhängigkeiten im Programmstück 3 P.
 - S1-S5 (\$s0)
 - S2-S3 (\$t0)
 - S3-S4 (\$t0)
 - S3-S7 (\$t0)
 - S4-S5 (\$s1)
 - S5-S6 (\$s1)
 - S6-S7 (\$s2)

4. Bei der folgenden Tabelle wird angenommen, dass Schreibvorgänge in den Registersatz im jeweiligen Taktzyklus bereits abgeschlossen werden.

3 P.

	IF	ID	EX	MEM	WB	\$s0	\$s1	\$t0
1	addi	(nop)	(nop)	(nop)	(nop)	0	0	0
2	lui	addi	(nop)	(nop)	(nop)	0	0	0
3	ori	lui	addi	(nop)	(nop)	0	0	0
4	lw	ori	lui	addi	(nop)	0	0	0
5	add	lw	ori	lui	addi	100	0	0
6	srl	add	lw	ori	lui	100	0	0x20140000
7	sw	srl	add	lw	ori	100	0	0x00001234
8	(nop)	sw	srl	add	lw	100	⊗	0x00001234
9	(nop)	(nop)	sw	srl	add	100	100	0x00001234
10	(nop)	(nop)	(nop)	sw	srl	100	100	0x00001234
11	(nop)	(nop)	(nop)	(nop)	sw	100	100	0x00001234

⊗: Datenwort an Adresse 0x00000004 (undefiniert)

5. DLX-Pipeline ohne Forwarding:

2 P.

S1
 S2
 NOP
 NOP
 S3
 NOP
 NOP
 S4
 NOP
 NOP
 S5
 NOP
 NOP
 S6
 NOP
 NOP
 S7

6. Modifizierte DLX-Pipeline mit Result Forwarding und Load Forwarding:

2 P.

S1
S2
S3
S4
NOP
S5
S6
S7

7. Berechnung der Ausführungszeit:

3 P.

- Sequentielle Ausführung: $7 \cdot 5 = 35$ Taktzyklen
- DLX-Pipeline ohne Forwarding: $17 + 4 = 21$ Taktzyklen
- DLX-Pipeline mit Forwarding: $8 + 4 = 12$ Taktzyklen