

Technische Grundlagen der Informatik			Test 3 26.01.2016 90 Minuten Gruppe A
Matrikelnr.	Nachname	Vorname	Unterschrift

Deckblatt sofort ausfüllen und unterschreiben!

Bitte deutlich und nur mit **Kugelschreiber** schreiben.  
Unleserliche Antworten werden nicht gewertet!

Geben Sie bei Rechenaufgaben den **Lösungsweg** an!

Buch, Mitschriften, Ausdrücke von Folien, Handys,  
Taschenrechner etc. sind nicht zugelassen!

Zusatzblätter werden nicht akzeptiert!

Bei **Ankreuzfragen** werden Minuspunkte auf Teilaufgaben  
übernommen. Das Minimum je Gesamtaufgabe beträgt 0  
Punkte.

1	[8]	[ ]
2	[8]	[ ]
3	[10]	[ ]
4	[10]	[ ]
5	[12]	[ ]
6	[12]	[ ]
7	[12]	[ ]
8	[10]	[ ]
9	[8]	[ ]
10	[10]	[ ]
Summe	[100]	[ ]

1. (8 Punkte) Gegeben ist eine Stackmaschine mit genau einem Register  $R$  und einem Stack. Die Stackmaschine kann ausschließlich folgende Operationen ausführen:

- *pop* entfernt den obersten Wert vom Stack und schreibt ihn in  $R$
- *push* kopiert den Inhalt von  $R$  auf die oberste Stackposition
- *add* entfernt die zwei obersten Elemente vom Stack und legt stattdessen deren Summe ab
- *goto [ADDR]* führt einen unbedingten Sprung zur Zeile/Sprungmarke [ADDR] aus

Geben Sie eine Befehlsfolge für diese Stackmaschine an, die die Folge  $2^1, 2^2, 2^3 \dots$  der Zweierpotenzen am Stack ablegt! Am Ende eines Schleifendurchlaufs soll die höchste Potenz immer an oberster Stelle des Stacks liegen. Zu Beginn ist der Stack leer und  $R$  ist mit 1 initialisiert.

```

:ADDR
push(R)
push(R)
add
pop(R)
goto ADDR

```

1  
2  
2  
4

2. (8 Punkte) Sie arbeiten mit einem Prozessor, der eine vierstufige Pipeline besitzt: Fetch (F), Decode (D), Execute (E) und Store (S).

Bedingt durch die Pipelinestruktur kann es zu *RAW Data Hazards* kommen, welche durch eine verzögerte Ausführung (*stall*) der lesenden Instruktion vermieden werden. Dabei wird die lesende Instruktion erst dann in Stufe D verarbeitet, wenn die schreibende Instruktion Stufe E abgeschlossen hat. Auf dem Prozessor wird folgendes Programm ausgeführt:

```
MULT R3, R4, R4 # R4 quadrieren, Resultat in R3
PUSH R3          # R3 auf Stack ablegen
SUB  R2, R3, R4  # R4 von R3 subtrahieren, Resultat in R2
SLL  R6, R5, 1   # Shift left von R5 um eine Stelle, Resultat in R6
ADD  R4, R2, R7   # R2 und R7 addieren, Resultat in R4
```

Zeichnen Sie die Belegung der Pipeline für das gegebene Programm unter der Voraussetzung, dass die Pipeline zu Beginn und am Ende leer ist! Verwenden Sie dabei die in der Übung präsentierte Notation. Die ersten beiden Zeilen sind bereits vorausgefüllt.

Zeit ↓	F	D	E	S
1	MULT			
2	PUSH	MULT		
3	SUB	(PUSH)	MULT	
4	SUB	(PUSH)		MULT
5	SLL	SUB	PUSH	
6	ADD	SLL	SUB	PUSH
7		ADD	SLL	SUB
8			ADD	SLL
9				ADD
10				
11				
12				
13				

Notizen:

2. (8 Punkte) Sie arbeiten mit einem Prozessor, der eine vierstufige Pipeline besitzt: Fetch (F), Decode (D), Execute (E) und Store (S).

Bedingt durch die Pipelinestruktur kann es zu *RAW Data Hazards* kommen, welche durch verzögerte Ausführung (*stall*) der lesenden Instruktion vermieden werden. Dabei wird die lesende Instruktion erst dann in Stufe D verarbeitet, wenn die schreibende Instruktion Stufe S abgeschlossen hat. Auf dem Prozessor wird folgendes Programm ausgeführt:

```
MULT R3, R4, R4  # R4 quadrieren, Resultat in R3
PUSH R3          # R3 auf Stack ablegen
SUB R2, R3, R4    # R4 von R3 subtrahieren, Resultat in R2
SLL R6, R5, 1     # Shift left von R5 um eine Stelle, Resultat in R6
ADD R4, R2, R7     # R2 und R7 addieren, Resultat in R4
```

Zeichnen Sie die Belegung der Pipeline für das gegebene Programm unter der Voraussetzung, dass die Pipeline zu Beginn und am Ende leer ist! Verwenden Sie dabei die in der Übung präsentierte Notation. Die ersten beiden Zeilen sind bereits vorausgefüllt.

Zeit ↓	F	D	E	S
1	MULT			
2	PUSH	MULT		
3	SUB	(PUSH)	MULT	
4	SUB	(PUSH)		MULT
5	SUB	PUSH		
6	SLL	SUB	PUSH	
7	ADD	SLL	SUB	PUSH
8		(ADD)	SLL	SUB
9		ADD		SLL
10			ADD	
11				ADD
12				
13				

Notizen:

$$10^3 \text{ s} = \text{ns}$$

3. (10 Punkte) Ein Prozessor besitzt eine fünfstufige Pipeline: *Fetch* (IF), *Decode* (ID), *Execute* (EXE), *Memory* (MEM) und *Write Back* (WB). Die Verarbeitungsdauer der einzelnen Stufen ist für alle Instruktionstypen gleich und nachfolgend angegeben:

IF	ID	EXE	MEM	WB	SUM
100 ns	400 ns	1000 ns	200 ns	300 ns	2000 ns

- (a) Geben Sie den theoretischen Durchsatz des Prozessors ohne Pipelining in MIPS an!

$$\frac{1}{2000} \cdot 1000 = 0,5 \text{ MIPS}$$

- (b) Geben Sie den theoretischen Durchsatz des Prozessors mit Pipelining in MIPS an!

$$\frac{1}{1000} \cdot 1000 = 1 \text{ MIPS}$$

- (c) Wie viele Mikrosekunden ( $\mu\text{s}$ ) beträgt die Durchlaufzeit einer Instruktion mit Pipelining, wenn während der Verarbeitung genau ein *stall* auftritt?

$$6 \mu\text{s} \quad 6 \cdot 1000 \text{ ns}$$

Um den theoretischen Durchsatz des Prozessors mit Pipeline zu verbessern, soll die *Execute*-Stufe durch eine der folgenden drei Varianten ersetzt werden:

EXE A
650 ns

EXE B1	EXE B2
600 ns	100 ns

EXE C1	EXE C2
500 ns	500 ns

- (d) Kreuzen Sie jene Variante an, die den theoretischen Durchsatz am meisten verbessert!

☐ Variante A

☐ Variante B1+B2

☒ Variante C1+C2

- (e) Berechnen Sie anhand der nachfolgenden Formel, um wie viel Prozent die gewählte Variante den theoretischen Durchsatz gegenüber Aufgabe (b) verbessert!

$$\text{Verbesserung [in \%]} = \left( \frac{\text{theoretischer Durchsatz nach Verbesserung [in MIPS]}}{\text{theoretischer Durchsatz vor Verbesserung [in MIPS]}} - 1 \right) \times 100$$

$$\frac{\frac{1}{500} \cdot 1000}{1} - 1 = 2 - 1 = 1 \Rightarrow 100\%$$

Notizen:



4. (10 Punkte) Für einen anfangs leeren Cache ist eine Sequenz von Adresszugriffen gegeben. Zwecks Vereinfachung wird keine Unterscheidung zwischen Lese- und Schreibzugriffen getroffen. Es handelt sich um einen *2-Way Set Associative Cache* mit 4 Sets. Pro Block können 4 Datenwörter gespeichert werden. Die Datenwortlänge ist 8 Bit und die Adresslänge beträgt 6 Bit. Die Adressierung erfolgt auf Datenwortebene.

(a) Tragen Sie in der nachfolgenden Tabelle ein, ob es sich bei dem Zugriff um einen *Hit* oder einen *Miss* handelt!

32 46 = 48 + 2 + 1

Adresse (dezimal)	Adresse (binär)	Hit/Miss	Set	Block
32	1 0 0 0 0 0	miss	0	0
5	0 0 0 1 0 1	miss	1	0
42	1 0 1 0 1 0	miss	2	0
15	0 0 1 1 1 1	miss	3	0
37	1 0 0 1 0 1	miss	1	1
34	1 0 0 0 1 0	hit	0	0
46	1 0 1 1 1 0	miss	3	1
13	0 0 1 1 0 1	hit	3	0
31	0 1 1 1 1 1	miss	3	0

- (b) Welche Speicherbereiche des Hauptspeichers können im Set 0 gespeichert werden? Geben Sie die Adressbereiche in der Notation  $mem[x-y]$  an, wobei  $x$  die erste und  $y$  die letzte dezimale Adresse eines Adressbereichs darstellen!

$mem[51-98], mem[3-0]$

- (c) Welche Ersetzungsstrategie wurde verwendet? Kreuzen Sie nur eine an!  
(korrekte Antwort: +2 Punkte, falsche Antwort: -2 Punkt, keine Antwort: 0 Punkte)

- ☐ Least Frequently Used (LFU)  
☐ Least Recently Used (LRU)  
☐ First In - First Out (FIFO)  
☒ Last In - First Out (LIFO)

9. (8 Punkte) Ordnen Sie die angegebenen Begriffe jener Schicht des OSI-7-Schichtenmodells zu, auf der sie operieren und benennen Sie die Schicht! Die erste Zeile ist bereits vorausgefüllt.

Begriff	Schicht	Bezeichnung
Netzwerkkabel	1	Physical Layer
Switch	2	Data-Link-Layer
Multi-Port Repeater	1	Physical-Layer
UDP	4	Transport-Layer
IPv6	3	Network-Layer

10. (10 Punkte) Ihr Computer hat folgende Netzwerkkonfiguration zugewiesen bekommen:

IP-Adresse: 192.168.5.110 128  
 Subnetmaske: 255.255.255.192 1  
 Default-Gateway: 192.168.5.65 00111111

- (a) Geben Sie das zugewiesene Netzwerk in CIDR-Notation an!

126

- (b) Geben Sie die Broadcast-Adresse des Netzwerks an!

IPv7 Sub  
 192.168.5.127

00111111  
 01101110  
 01111111

69+32=96  
 $\frac{+}{109}$

- (c) Welche in der Vorlesung besprochene Funktionalität muss Ihr Default-Gateway unterstützen, damit Sie Pakete an die Adresse 120.130.35.76 schicken können?

NAT (Network Address Translation)

- (d) Können Sie die IP-Adresse 127.0.0.1 pingen? Was bezeichnet diese Adresse?

Ja  
 local host (am selbst)

Notizen: