

Technische Grundlagen der Informatik			Test 3 20.01.2017 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	[12]	[]
2	[12]	[]
3	[10]	[]
4	[8]	[]
5	[12]	[]
6	[10]	[]
7	[8]	[]
8	[8]	[]
9	[10]	[]
10	[10]	[]
Summe	[100]	[]

1. (12 Punkte) Kreuzen Sie an, ob es sich um wahre oder falsche Aussagen handelt!
(richtig: +2 Punkte, falsch: -2 Punkte, keine Antwort: 0 Punkte; Minimum: 0 Punkte)

wahr falsch

- ☐ ☐ Eine virtuelle Seite muss größer als ein Frame im physischen Speicher sein.
- ☒ ☐ Die Abkürzung DMA steht für Direct Memory Access.
- ☒ ☐ Angenommen, ein Stack ist leer und der Stackpointer zeigt auf die Adresse 0xFFFF. Dann ist die Instruktion `memory[(SP)-] ← R2` gleichzusetzen mit einem `push(R2)`.
- ☐ ☒ Beim Fully Associative Cache gibt es zu jeder Speicheradresse eine genau festgelegte Position im Cache.
- ☐ ☒ Der Einsatz einer Pipeline verbessert die Ausführungszeit eines einzelnen Befehls.
- ☒ ☐ Ein Direct Mapped Cache benötigt keine Ersetzungsstrategie (wie z.B. LRU), da dessen Assoziativität gleich 1 ist.

3. (10 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. Analog zur Übung gilt, dass die lesende Instruktion erst dann in Stufe D verarbeitet wird, wenn die schreibende Instruktion Stufe S abgeschlossen hat. Auf dem Prozessor wird folgendes Programm ausgeführt:

```

DIV  R4, R2, R1 # R2 durch R1 dividieren, Resultat in R4
MOV  R3, 1      # 1 nach R3 kopieren
ADD  R5, R4, R3 # R4 und R3 addieren, Resultat in R5
MULT R1, R5, R6 # R5 und R6 multiplizieren, Resultat in R1
PUSH R1         # lege den Inhalt von R1 am Stack ab

```

Zeichnen Sie die Belegung der Pipeline für das gegebene Programm unter der Annahme, dass die Pipeline am Beginn und am Ende leer ist!

Zeit ↓	F	D	E	S
1	DIV			
2	MOV	DIV		
3	ADD	MOV	DIV	
4	MULT	(ADD)	MOV	DIV
5	MULT	(ADD)		MOV
6	MULT	ADD		
7	PUSH	(MULT)	ADD	
8	PUSH	(MULT)		ADD
9	PUSH	MULT		
10		(PUSH)	MULT	
11		(PUSH)		MULT
12		PUSH		
13			PUSH	
14				PUSH
15				
16				
17				

4. (8 Punkte) Stack und Register eines Prozessors sind wie folgt initialisiert, wobei alle Werte hexadezimal angegeben sind:

	Stack		Register
...			0 R1
FFFC			0 R2
FFFD	1		0 R3
FFFE	3		FFFC SP
FFFF	8		

Auf dem Prozessor werden die unten links angeführten acht Befehle von oben nach unten ausgeführt. Tragen Sie die Werte auf dem Stack, der Register und des Stackpointers in hexadezimaler Form jeweils nach Ausführung der beiden Instruktionen ein!

Geben Sie in eckigen Klammern [] auch jene Daten auf dem Stack an, die durch vorangegangene pop()-Befehle bereits ungültig sind, die aber noch nicht überschrieben wurden!

		Stack		Register
pop(R1)	FFFC			1 R1
pop(R2)	FFFD	[1]		3 R2
	FFFE	[3]		0 R3
	FFFF	8		FFFF SP

		Stack		Register
R1 ← R1 + R2	FFFC			4 R1
pop(R3)	FFFD	[1]		3 R2
	FFFE	[3]		8 R3
	FFFF	[8]		ffff SP

		Stack		Register
push(R1)	FFFC			4 R1
rsh(R3)	FFFD	[1]		3 R2
	FFFE	[3]		4 R3
	FFFF	4		ffff SP

		Stack		Register
R2 ← R2 + R3	FFFC			4 R1
push(R2)	FFFD	[1]		7 R2
	FFFE	7		4 R3
	FFFF	4		ffff SP

7. (8 Punkte) Kreuzen Sie an, ob es sich um wahre oder falsche Aussagen handelt!
(richtig: +2 Punkte, falsch: -2 Punkte, keine Antwort: 0 Punkte; Minimum: 0 Punkte)

wahr falsch



UDP ist ein verbindungsloses Protokoll auf Layer 4.



Ein NOR-Flash kann schneller gelesen werden als ein NAND-Flash.



Bei einer Vollduplex-Übertragung können Daten abwechselnd, aber nicht gleichzeitig, in beide Richtungen übertragen werden.



Eine Bridge verbindet immer gleich schnelle Datenbusse im PC miteinander.

unterschiedlich

8. (8 Punkte) Gegeben ist die folgende IPv6-Adresse in minimaler Notation:

BAD:0:BEEF::D5E:FE/120

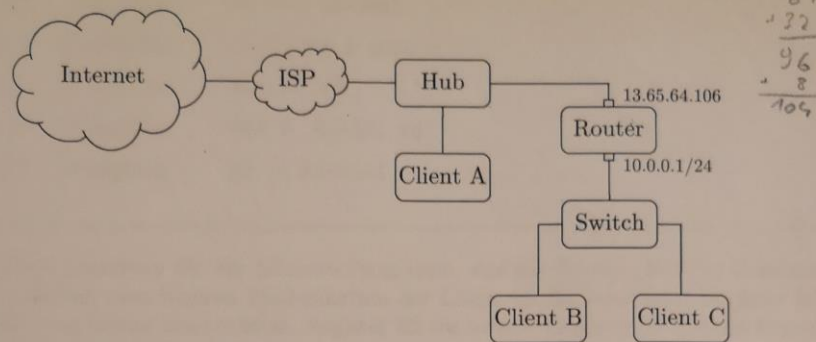
(a) Geben Sie die vollständige Adresse mit allen führenden Nullen an!

(b) Wandeln Sie die ursprünglich gegebene Adresse in die gemischte Schreibweise um, bei der die letzten 4 Bytes in Dezimalnotation angegeben sind!

(c) Wie viele Adressen sind im gegebenen Netzwerk verfügbar?

(d) Wie viele Subnetze mit je 8 Adressen könnten im gegebenen Netzwerk erstellt werden?

9. (10 Punkte) Sie bekommen von Ihrem ISP (Internet Service Provider) ein /29 Netzwerk zugewiesen. Der Default-Gateway 13.65.64.105 befindet sich im zugewiesenen Netzwerk aber physisch beim ISP. Im Weiteren bauen Sie folgende Netzwerkstruktur auf:



- (a) Wie lauten die Netzwerk- und die Broadcast-Adresse des zugewiesenen Netzwerkes?

N: 10 x 24 = 13.65.64.104
 B: 10 x 24 = 13.65.64.111
 Sub: 1111 1111 1111 1111 1111 1111 1111 1111
 00000111
 01101010
 01101111
 64 21
 13 + 32 + 64
 96 111
 24

- (b) Weisen sie dem Client A eine Netzwerkkonfiguration (IP-Adresse, Subnetmask, Gateway) zu, so dass dieser mit dem Internet kommunizieren kann!

13.65.64.108, 255.255.255.0, 13.65.64.105

- (c) Sie wollen einen weiteren Client (beispielsweise einen Webserver) in Ihrem zugewiesenen Netzwerk betreiben. An welches Gerät in Ihrer Netzwerkstruktur sollte dieser angeschlossen werden, so dass eine direkte Kommunikation (ohne Portforwarding) mit dem Internet möglich ist?

Router

- (d) Jemand hat sich Zugang zu Ihrem Netzwerk verschafft und ein Gerät am Hub angeschlossen. Wieso kann dieses Gerät Informationen, die zwischen Client B und dem Internet fließen, mitloggen?

weil Hub an alle Teilnehmer schickt
 Daten

- (e) Der Router in ihrem Netzwerk unterstützt NAT und kein Portforwarding. Kann der Client A eine Verbindung zu Client C aufbauen? Begründen Sie Ihre Antwort!

Nein weil öffentlich mit ohne Portforwarding in Privates Netzwerk rein kommen!

9. (6 Punkte) Analysieren Sie die folgenden Micro16-Instruktionen und kreuzen Sie korrekte an!
(korrekte Antwort: +1 Punkt, falsche Antwort: -1 Punkt, keine Antwort: 0 Punkte)

gültig <input type="radio"/>	ungültig <input type="radio"/>	$(R1 \leftarrow R9 \wedge 1); \text{ if } N \text{ goto } 127$
gültig <input type="radio"/>	ungültig <input type="radio"/>	$R2 \leftarrow \neg \text{lsh}(R6)$
gültig <input type="radio"/>	ungültig <input type="radio"/>	$(\neg 1); \text{ if } Z \text{ goto } 9$
gültig <input type="radio"/>	ungültig <input type="radio"/>	$MAR \leftarrow MBR$
gültig <input type="radio"/>	ungültig <input type="radio"/>	$MAR \leftarrow R1 + R2; \text{ rd}$
gültig <input type="radio"/>	ungültig <input type="radio"/>	$RO \leftarrow R1 + R2 + 1$

10. (12 Punkte) Schreiben Sie ein Micro16-Programm, das die Anzahl der übereinstimmenden Einsen zwischen zwei binären Zeichenketten der Länge 16 Bit berechnet! Register R1 beinhaltet die erste binäre Zeichenkette, Register R2 die zweite. Register R7 soll das Ergebnis als Zweierkomplementzahl beinhalten.

Beispiele:

$R1 = 00000000 \ 0010110, R2 = 00000000 \ 0010100 \rightarrow R7 = 00000000 \ 00000010$

$R1 = 11111111 \ 0010110, R2 = 00000000 \ 1111111 \rightarrow R7 = 00000000 \ 00000011$

i loop

(R1); if Z goto end

(R2); if Z goto end

(R1 & R2); if Z goto none

(R2 & R1); if Z goto none

R7 ← R7 + 1

:none

R1 ← RSH(R1)

R2 ← RSH(R2)

goto loop

end

Notizen:

5. (12 Punkte) Gegeben ist ein Prozessor mit einer Adresslänge von 8 Bit sowie einer Datenwortlänge von 1 Byte. Der integrierte *Direct Mapped Cache* besitzt 4 Blöcke und speichert 8 Byte pro Block. Es wird auf Datenwortebene adressiert. Bei einem schreibenden Zugriff wird bei einem *Hit* das Verfahren *Copy-Back* verwendet, bei einem *Miss* das Verfahren *Write-Around*.

(a) Geben Sie die Längen von Tag, Index und Offset an!

Tag: $8 - 5 = 3$ Bit

Index: 2 Bit

Offset: 3 Bit

Auf dem Prozessor werden folgende Speicherzugriffe ausgeführt:

$r(0x23) - w(0x5F) - r(0xC8) - r(0x1C) - w(0x1D) - r(0x20) - r(0x1E) - r(0x39)$

Die Verwaltungsdaten ändern sich dabei wie folgt:

	Speicherzugriff				Block 0			Block 1			Block 2			Block 3		
	Adr.	r/w	h/m	Adr. bin.	V	D	Tag	V	D	Tag	V	D	Tag	V	D	Tag
1	0x23	r	m	00100011	1	0	0x1	0	0	0x0	0	0	0x0	0	0	0x0
2	0x5F	w	m	01011111	1	0	0x1	0	0	0x0	0	0	0x0	0	0	0x0
3	0xC8	r	m	11001000	1	0	0x1	1	0	0x6	0	0	0x0	0	0	0x0
4	0x1C	r	m	00011100	1	0	0x1	1	0	0x6	0	0	0x0	1	0	0x0
5	0x1D	w	h	00011101	1	0	0x1	1	0	0x6	0	0	0x0	1	1	0x0
6	0x20	r	h	00100000	1	0	0x1	1	0	0x6	0	0	0x0	1	1	0x0
7	0x1E	r	h	00011110	1	0	0x1	1	0	0x6	0	0	0x0	1	1	0x0
8	0x39	r	m	00111001	1	0	0x1	1	0	0x6	0	0	0x0	1	0	0x1
9	0x0D	w	m	00001101												
10	0x27	w	h	00100111	1	1	0x1									

(b) Bei welchem Speicherzugriff wurde *Copy-Back* verwendet?

5

(c) Bei welchem Speicherzugriff wurde zum ersten Mal in den Hauptspeicher geschrieben?

2

(d) Bei welchem Speicherzugriff wurde zum ersten Mal ein gültiger Block im Cache ersetzt?

8

(e) Ergänzen Sie in obiger Tabelle die beiden folgenden Speicherzugriffe:

$w(0x0D) - w(0x27)$

Hinweis: Bei den Blöcken müssen Sie nur Felder befüllen, wo sich eine Änderung ergibt!

2. (12 Punkte) Gegeben ist der folgende Ausschnitt aus dem Speicher eines Computers, wobei alle Adressen und Konstanten hexadezimal angeschrieben sind:

Adresse	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF
Wert	A5	39	69	32	FD	28	11	0	41	30	A	AF	23	B1	5A	C7

Auf diesem Computer wird folgende Befehlsfolge ausgeführt:

- 1 $\text{memory}[A5] \leftarrow R1$
- 2 $R0 \leftarrow \text{memory}[(R2)+]$
- 3 $\text{memory}[R2+5] \leftarrow \text{memory}[AD]$
- 4 $R1 \leftarrow \text{memory}[\text{memory}[AB]]$
- 5 $\text{memory}[-(R2)] \leftarrow R1$
- 6 $R0 \leftarrow \text{memory}[R1-1B]$

Tragen Sie den Inhalt der Register *nach* jedem Befehl in die folgende Tabelle ein! Geben Sie weiters Art und Adresse der Speicherzugriffe an! Beachten Sie die Initialisierung der Register. Die erste Zeile ist bereits vorausgefüllt.

Befehl	R0	R1	R2	Speicherzugriff(e)
0	0	20	A1	
1	0	20	A1	wr(A5)
2	39	20	A2	rd(A1)
3	39	20	A2	rd(AD) wr(A7)
4	39	C7	A2	rd(AF)
5	39	C7	A1	wr(A1)
6	23	C7	A1	rd(AC)

Platz für Notizen:

$$\begin{array}{r} 12 \cdot 16 \\ 12 \\ \hline 192 \end{array}$$

$$16 + 11 = 27$$

$$\begin{array}{r} 192 + 7 = 199 \\ - 27 \\ \hline 172 \end{array}$$

$$\begin{array}{r} 172 \cdot 16 = 10 \\ 10 \cdot 16 = 0, \end{array}$$

$$\begin{array}{r} 172 : 16 = 10 \\ 10 \\ \hline 12 \\ 12 \cdot C \\ 10 = 4 \end{array}$$