

Technische Grundlagen der Informatik

Test 3
26.6.2012
60 Minuten
Gruppe C

Matrikelnr.	Nachname	Vorname	Unterschrift

1	[12]
2	[13]
3	[10]
4	[15]
5	[10]
6	[15]
7	[10]
8	[5]
9	[10]
Summe	[100]

Deckblatt sofort ausfüllen!

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

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

Zusatzblätter werden nicht akzeptiert!

1. (12 Punkte) Berechnen Sie für folgende IP-Adressen in CIDR-Notation die Netzadresse des Zielnetzes und geben Sie die Anzahl der adressierbaren Hosts an!

(a) 192.168.1.3/21

Netzadresse (3 Punkte):

Anzahl Hosts (2 Punkte):

(b) 192:168:1::3/21

Netzadresse (5 Punkte):

Anzahl Hosts (2 Punkte):

2. (13 Punkte) Sie arbeiten mit einem Prozessor, der eine vierstufige Pipeline besitzt: Fetch Instruction (F), Decode Instruction (D), Execute (E), Store Result (S). Bedingt durch die Pipeline-Struktur kann es zu *RAW (Read After Write) Data Hazards* kommen, welche durch verzögerte Ausführung des abhängigen Befehls (*stall*) vermieden werden. Dabei wird eine Instruktion erst dann in Stufe D verarbeitet, wenn die abhängige Instruktion Stufe S abgeschlossen hat.

Auf diesem Prozessor wird folgendes Programm ausgeführt:

```

i1  ADD  R1, R2, R2      # R1 = R2 + R2
i2  PUSH R1              # R1 auf den Stack legen
i3  MULT R2, R4, R4      # R4 quadrieren, Resultat in R2
i4  SLL  R1, R4, 1        # Shift left von R4, Resultat in R1
i5  SUB  R4, R4, R1       # R1 von R4 abziehen, Resultat in R4
i6  POP  R2              # oberstes Stackelement in R2 lesen
i7  DIV  R2, R5, R1       # R5 durch R1 dividieren, Resultat in R2

```

- (a) (7 Punkte) Zeichnen Sie die Belegung der Pipeline für das gegebene Programm unter der Annahme, dass die Pipeline zu Beginn und am Ende leer ist.

Zeit ↓	F	D	E	S
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				

- (b) (2 Punkte) Wieviele Takte braucht die CPU um diese Befehlsfolge abzuarbeiten?

- (c) (4 Punkte) Welchem Durchsatz (Instruktionen pro Sekunde) entspricht das, wenn jede Pipelinestufe 1 ns benötigt?

4. (15 Punkte) Ein System arbeitet mit 4096 Byte großen Pages und Frames, die kleinste adressierbare Einheit ist 1 Byte. Zu Beginn werden die folgenden (virtuellen) Adresszugriffe von links nach rechts abgearbeitet, wobei Frames in aufsteigender Reihenfolge (bei 00 beginnend) beschrieben werden und als Ersetzungsstrategie *First In – First Out (FIFO)* verwendet wird:

$(7CAD)_{16}$ $(3252)_{16}$ $(4341)_{16}$ $(02A5)_{16}$ $(534D)_{16}$

Die resultierende Page-Table lautet:

Page-NR	Frame-NR	Present-Bit
000	11	1
001	00	0
010	00	0
011	01	1
100	10	1
101	00	1
110	00	0
111	00	0

- (a) (3 Punkte) Wieviel Byte *physischer* Speicher sind im System vorhanden?
- (b) (3 Punkte) Auf welche Page greift die Adresse $(534D)_{16}$ zu?
- (c) (3 Punkte) Wie lautet die entsprechende *physische* Adresse zur (virtuellen) Adresse $(534D)_{16}$?
- (d) (6 Punkte) Wie *verändert* sich die Page-Table durch die folgenden beiden Adresszugriffe (Ersetzungsstrategie = FIFO):

$(32EA)_{16}$ – $(2CAC)_{16}$

Page-NR	Frame-NR	Present-Bit
000		
001		
010		
011		
100		
101		
110		
111		

Hinweis: Sie brauchen nur jene Zeilen eintragen, die sich gegenüber der obigen Darstellung der Page-Table verändern.

5. (10 Punkte) Gegeben ist der folgende Ausschnitt aus dem Speicher eines Computers, wobei alle Adressen und Konstanten hexadezimal angeschrieben sind:

Adresse	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF
Wert	F1	10	D9	21	4F	A2	F0	0	12	1E	99	78	25	11	10	16

Auf diesem Computer wird folgende Befehlsfolge ausgeführt:

```

1           SP ← FF
2           R1 ← F5
3   memory[FA] ← R1
4   memory[-(SP)] ← R1
5           R1 ← memory[memory[F6]]
6           R0 ← memory[(SP)+]
7           R1 ← memory[(R0)]
8   memory[R0 + 6] ← R1

```

Tragen Sie den Inhalt der Register *nach* jedem Befehl in die folgende Tabelle ein. Geben Sie weiters die Speicheradresse(n) an, falls ein Speicherzugriff erfolgt.

Befehl	SP	R0	R1	Speicheradresse(n)
1				
2				
3				
4				
5				
6				
7				
8				

6. (15 Punkte) Ein Prozessor mit einer Adresslänge von 24 Bit und einer Datenwortlänge von 8 Bit hat einen 16 KByte (1 KByte $\hat{=}$ 1024 Byte) großen 4-way Set-Associative Cache mit einer Blockgröße von 16 Byte.

(a) (3 Punkte) Aus wievielen Cache-Sets ("Zeilen") besteht dieser Cache?

(b) (3 Punkte) Berechnen Sie für den gegebenen Cache die Längen von Tag, Offset und Index in Bit.

Tag-Länge:

Index-Länge:

Offset-Länge:

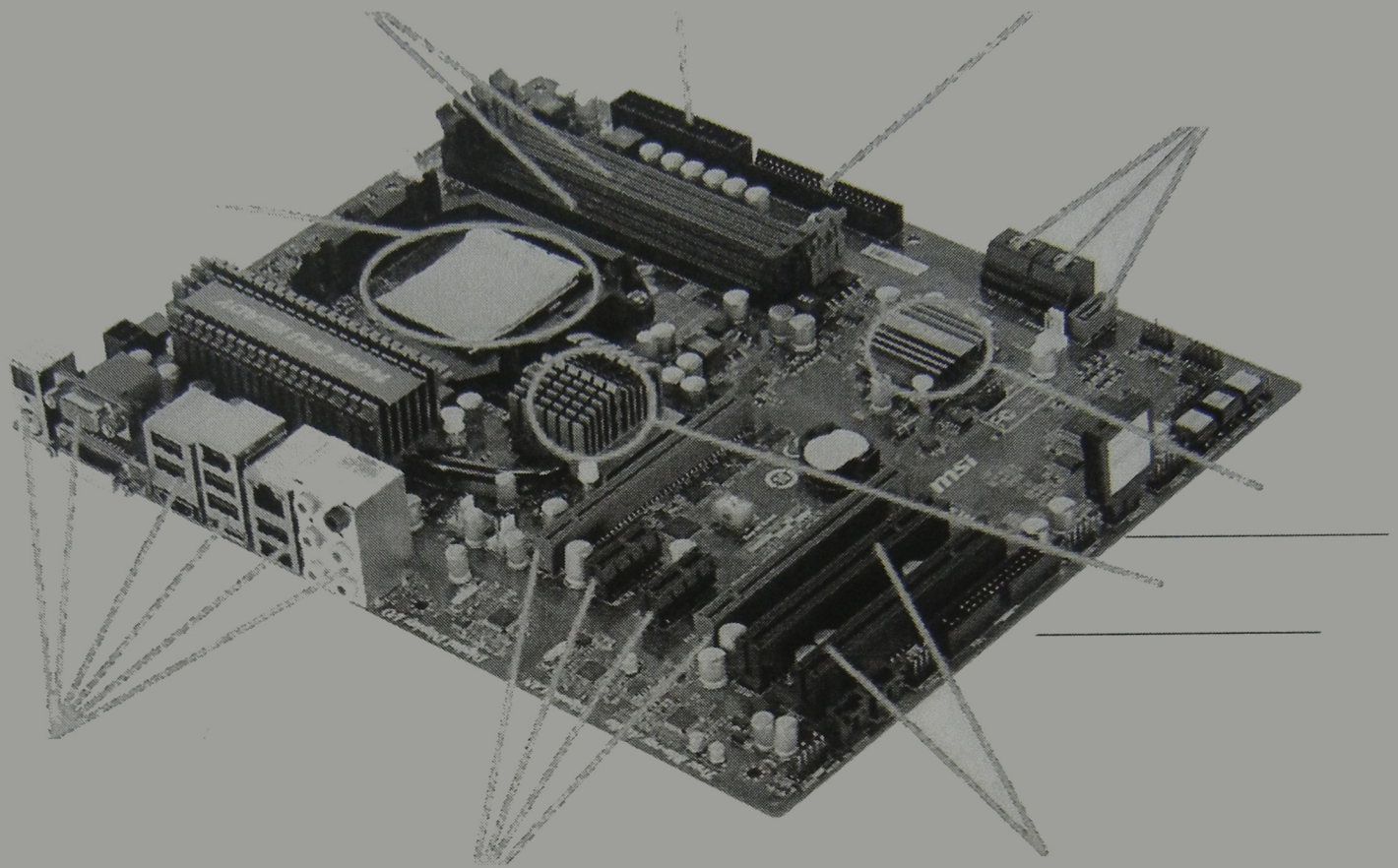
(c) (4 Punkte) Angenommen, das Cache-Set 8C ist vor jedem der unten angegebenen Lesezugriffe folgendermaßen belegt:

Tag	Valid	Dirty
DE0	1	0
003	1	1
3FF	1	0
7E8	0	0

Welche Zugriffe führen zu einem *miss*, welche zu einem *hit*? Kreuzen Sie entsprechend an!
 (korrekte Antwort: +1 Punkt, falsche Antwort: -1 Punkt, keine Antwort: 0 Punkte)

(e) (3 Punkte) Wieviele Adressen des Hauptspeichers sind dem selben Cache-Set zugeordnet?

7. (10 Punkte) Beschriften Sie die Komponenten des Mainboards in nachfolgender Abbildung:



8. (5 Punkte) Erklären Sie kurz den Begriff *Polling* anhand der USB 2.0-Schnittstelle.

9. (10 Punkte) Ein Messwert, der im Bereich $(-195)_{10}$ bis $(+60)_{10}$ liegt, soll über ein Netzwerk als 8 Bit Zahl in *Exzessdarstellung* übertragen werden.

(a) (4 Punkte) Geben Sie einen geeigneten Exzess e in Binärdarstellung an.

(b) (3 Punkte) Geben Sie die binäre Codierung der Zahl $(0)_{10}$ in dieser Darstellung an.

(c) (3 Punkte) Geben Sie die Codierung der Zahl $(-191)_{10}$ in dieser Darstellung an.