

---

## Aufgabe 1: Allgemein

8 Punkte

---

Welche der folgenden Aussagen sind wahr, welche falsch? Kreuzen Sie an.

Punktabzug bei falschen Antworten!

### 1. Prozessorarchitektur

2 Punkte

wahr falsch

- Maschineninstruktionen für RISC-CPU's sind meist gleich groß.
- Die Code-Dichte ist bei CISC-CPU's im Allgemeinen höher als bei RISC-CPU's.
- Arithmetische Befehle einer stackbasierten CPU besitzen *einen* expliziten Operanden.
- Bei einer akkumulator-basierten CPU muss bei arithmetischen Befehlen das Zielregister angegeben werden.

### 2. Ein-/Ausgabe

2 Punkte

wahr falsch

- Eine CISC-CPU kann nur mithilfe expliziter Ein-/Ausgabe-Befehle auf Peripherie zugreifen.
- Interrupts dienen der synchronen Unterbrechung des Prozessors.
- Ein DMA-Controller sendet bei Beendigung des Auftrags einen Interrupt an die CPU.
- Unterbrechungsgesteuerte Ein-/Ausgabe eignet sich besonders für große Datenmengen.

### 3. Assemblerprogrammierung

2 Punkte

wahr falsch

- Parameterübergabe über den Stack ist langsamer als über Register.
- Die Verwendung von *caller-save*-Registern in der aufgerufenen Funktionen erfordert zusätzliche Vorkehrungen.
- Rückgabewerte können über den Stack übergeben werden.
- Auf dem Stack übergebene Parameter müssen von der aufgerufenen Funktion entfernt werden.

### 4. Speicherverwaltung

2 Punkte

wahr falsch

- Bei Segmentierung tritt keine interne Fragmentierung auf.
- Mit Segmentierung ist keine Auslagerung von Arbeitsspeichereinhalten auf der Festplatte (Swapping) möglich.
- Der Translation Lookaside Buffer (TLB) übersetzt physikalische in virtuelle Adressen.
- Paging und Segmentierung schließen sich gegenseitig aus.

---

**Aufgabe 2: Assembler****14 Punkte**

---

Gegeben sei die folgende C-Funktion zur Sortierung eines Arrays mithilfe des Selectionsort:

```
1 void selection_sort(int *array, int size)
2 {
3     for (int i = 0; i < size; ++i) {
4         int idx = i;
5         int min = array[i];
6         for (int j = i + 1; j < size; ++j) {
7             int num = array[j];
8             if (num < min) {
9                 min = num;
10                idx = j;
11            }
12        }
13        int tmp = array[i];
14        array[i] = array[idx];
15        array[idx] = tmp;
16    }
17 }
```

1. Formen Sie das Programm so in if-goto-Darstellung um, sodass es sich möglichst leicht in Assembler übersetzen lässt.

Ziehen Sie hierbei außerdem die Definition der Variablen an den Anfang der Funktion.

8 Punkte

Fortsetzung von Aufgabe 2.1.

2. Wie hoch ist der Speicherbedarf für den Stack-Frame der Funktion unter folgenden Voraussetzungen?

- `int` und `int *` sind jeweils 32 Bits breit
- Lokale Variablen werden auf dem Stack abgelegt
- Arrayzugriffe benötigen keinen zusätzlichen Speicher auf dem Stack
- Es wird kein Frame-Pointer verwendet
- Es werden keine Optimierungen durchgeführt

Begründen Sie Ihre Antwort kurz.

2 Punkte

3. Nennen und erklären Sie zwei Möglichkeiten, um den erforderlichen Speicher des Stack-Frames zu reduzieren.

2 Punkte

4. Welche Adressierungsart des Assemblers verwendet man sinnvollerweise für das Laden von Arrayelementen aus dem Speicher (z.B. in Zeile 5 in obiger Funktion)?

Erklären Sie diese Adressierungsart kurz.

2 Punkte

---

## Aufgabe 3: Speicherverwaltung

8 Punkte

---

Ab der Adresse 0xCBB8 liege das 64-Bit-Wort 0xBA AD CA FE F0 OD BE AF im Speicher.

1. Stimmt das Alignment dieses Datenwortes? (Keine Begründung) 1 Punkt

2. Welchen Nachteil haben Zugriffe auf ein nicht korrekt ausgerichtetes Datenwort, verglichen mit einem korrekt ausgerichtetem?

Warum ist dies so? 2 Punkte

3. Wie liegt das Datenwort bei Little-Endian-Bytereihenfolge im Speicher?

Schreiben Sie den Inhalt jedes Bytes in das dazugehörige Feld. 2 Punkte

Adresse:	0xCBB8	0xCBB9	0xCBBA	0xCBBB	0xCBBC	0xCBBD	0xCBBE	0xCBBF
Inhalt:								

4. Was ist ein Burst-Zugriff auf den Arbeitsspeicher?

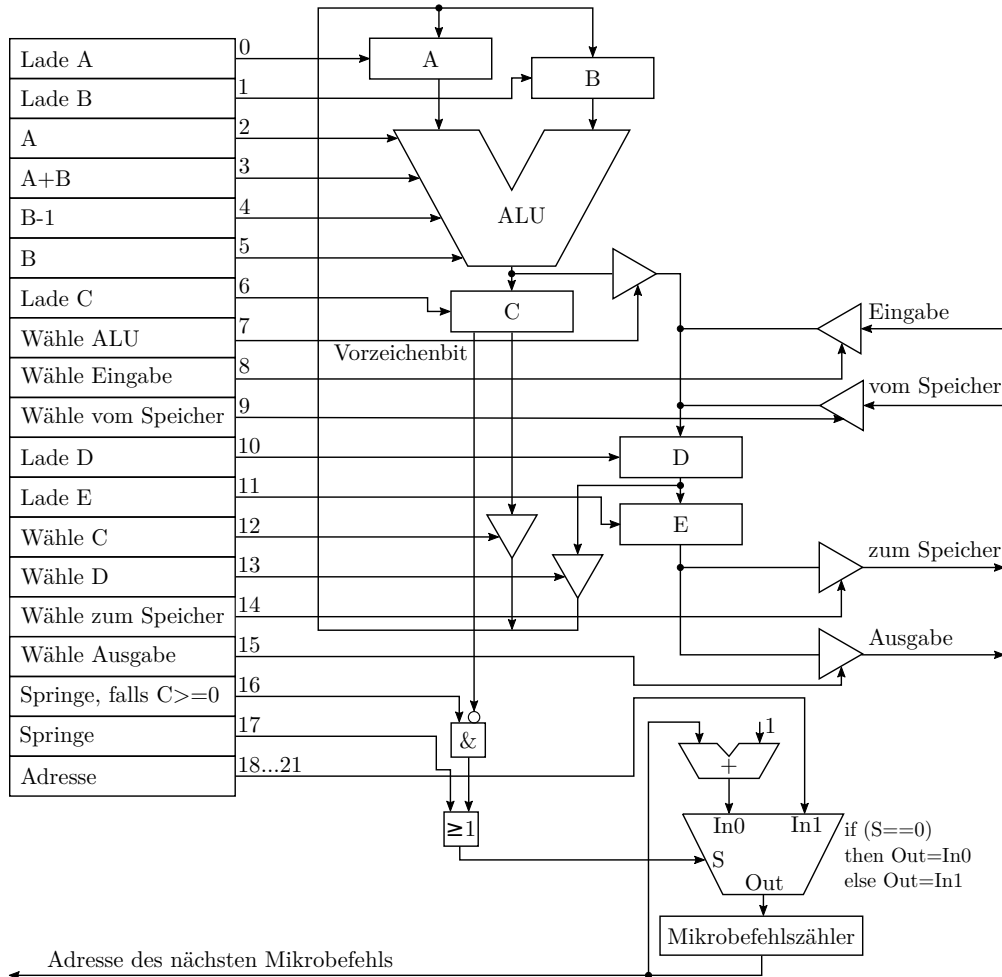
Wie kann man dadurch eine Beschleunigung beim Zugriff auf obiges Datenwort erhalten, wenn der Datenbus 32 Bits breit ist?

Gehen Sie darauf ein, an welcher Stelle Zeit gespart wird. 3 Punkte

# Aufgabe 4: Mikroprogrammierung

14 Punkte

Gegeben sei folgender Teil eines mikroprogrammierten Prozessors:



Die obige CPU soll nun um den Makrobefehl  $\text{Reg}[A] = \sum_{n=0}^N$  erweitert werden. Der Parameter  $N$  sei echt größer Null und soll einmalig von der Eingabe gelesen werden. Bei Eingabe von  $N = 3$  soll also bspw. das Ergebnis 6 ins Register A geschrieben werden.

Alle Register haben zu Beginn den Wert 0 und dürfen überschrieben werden. Eine Fehlerbehandlung ist nicht nötig.

- Überlegen Sie sich den Ablauf in Pseudocode, der sich an den möglichen Operationen orientiert. 2 Punkte



2. Schreiben Sie ein Mikroprogramm zur Realisierung des beschriebenen Befehls.

Verwenden Sie die folgende Tabelle und tragen Sie jeweils auch die Bedeutung (z. B. D → E) in die Spalte „Erklärung“ ein.

Leer gelassene Steuerleitungen entsprechen dem Wert 0. Das niederwertigste Bit des Sprungziels entspricht Steuerleitung 21. Vermeiden Sie unnötige Befehle und Sprünge.

(Die Tabellenlänge entspricht nicht der erwarteten Mikroprogrammlänge!) 10 Punkte

Adr.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	Erklärung		
0																									
1																									
2																									
3																									
4																									
5																									
6																									
7																									
8																									
9																									
10																									
11																									
12																									
13																									
14																									
15																									

3. Erklären Sie den Unterschied zwischen horizontaler und vertikaler Mikroprogrammierung und nennen Sie jeweils einen Vorteil. 2 Punkte

---

## Aufgabe 5: Parallelverarbeitung

13 Punkte

---

1. Leiten Sie rechnerisch den asymptotischen Speedup einer Pipeline-Architektur mit  $s$  Pipelinestufen her. 2 Punkte

2. Unterscheiden sich Hochsprachenprogramme für VLIW-basierte und superskalare Architekturen? (Keine Begründung)

Von wem wird jeweils die Parallelisierung umgesetzt? 2 Punkte

3. Warum ist eine sehr hohe Anzahl an Pipelinestufen in der Praxis nicht sinnvoll?

Nennen Sie die entsprechenden Fachbegriffe und erklären Sie kurz deren Auswirkung auf die Pipeline! 4 Punkte

4. Neben simultanem Multithreading wurden zwei weitere Arten behandelt. Nennen Sie diese und geben sie bei beiden Arten an, wie viele Threadkontexte, bestehend aus Befehlszähler, Universalregister, Flags-Register, gleichzeitig aktiv sind. (Keine Begründung) 2 Punkte

5. Unter welchen Umständen kann eine CPU mit simultanem Multithreading zwei Threads schneller ausführen als eine CPU ohne? 1 Punkt

6. Wann könnte eine CPU mit simultanem Multithreading zwei Threads schneller ausführen als eine Dual-Core-CPU? 2 Punkte

---

## Aufgabe 6: Cache

**15 Punkte**

---

1. Was versteht man unter räumlicher und zeitlicher Lokalität typischer Programme? 2 Punkte

2. Welche Rolle spielt die Ersetzungsstrategie bei direktabbildenden Caches im Vergleich zu vollassoziativen? 2 Punkte

3. Eine CPU besitzt einen 64 KiB großen direktabbildenden Cache mit einer Blockgröße von 64 Byte. Der Cache ist zu Beginn leer. Es werden insgesamt mehr Daten adressiert, als im Cache gespeichert werden können. Pro Hauptspeicher-Block erfolgt nur ein einziger Zugriff, es gibt also keine Cache-Hits.

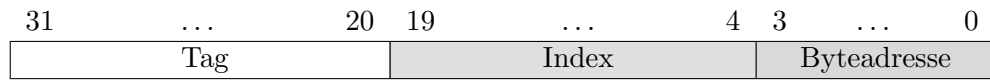
a) Beim wievielten Speicherzugriff findet frühestens eine Verdrängung aus dem Cache statt?

b) Beim wievielten findet spätestens eine Verdrängung statt?

c) Wie nennen sich die Misses, die bei obigen Zugriffen auftreten?

3 Punkte

4. Gegeben sei folgende Adressaufteilung:



Welche Organisationsformen liegen bei den folgenden Cachegrößen (nur Nutzdaten, keine Verwaltungsinformationen) vor?

Begründen Sie Ihre Antwort rechnerisch!

a)  $2^{20}$  Byte

b) 4 MiB

4 Punkte

5. Geben Sie die Adressaufteilung in Tag, Index (wo zutreffend) und Byteadresse für folgende Cache-Konfigurationen auf einem 32-Bit-Rechner an:

a) 4 KiB Nutzdaten,  $2^4$  Byte pro Block, 2-fach assoziativ

b) 64 KiB Nutzdaten, 1024 Blöcke, vollassoziativ

4 Punkte

---

## Aufgabe 7: Paging

18 Punkte

---

Eine fiktive CPU bietet eine Memory-Management-Unit mit den Eigenschaften:

- Zweistufige Adresstabellen
- Je 1024 Einträge zu je 4 Bytes in den Tabellen
- Seiten zu je 4 KiB Größe

Die einzelnen Bits der Tabelleneinträge haben folgende Bedeutung:

31-12	11-5	4	3	2	1	0
höherwertige Adressbits der physikalischen Adresse	unbenutzt (immer 0)	Access	Cache-Disable	Execute-Enable	Write-Enable	Present

Bits 1-4 haben in der ersten Tabellenstufe keine Bedeutung und müssen genullt werden.

Der Adressraum soll wie folgt aufgebaut sein:

	physikalischer Adressraum	virtueller Adressraum	Berechtigungen
ROM1	0x00 00 10 00 - 0x00 00 2F FF	0x00 04 00 00 - 0x00 04 1F FF	ausführbar, cachebar
ROM2	0x00 00 30 00 - 0x00 00 BF FF	nicht abgebildet	
RAM1	0x00 01 00 00 - 0x00 01 2F FF	0x00 05 00 00 - 0x00 05 2F FF	schreibbar, cachebar
RAM2	0x00 01 30 00 - 0x00 01 4F FF	0xFF FF E0 00 - 0xFF FF FF FF	schreibbar, cachebar
UART	0xF0 00 10 00 - 0xF0 00 1F FF	0xF0 00 10 00 - 0xF0 00 1F FF	schreibbar

Für die Seitentabellen sind die nicht im virtuellen Adressraum abgebildeten Seitenrahmen vorgesehen. Das `Page Directory Basis Register` enthält den Wert `0x3000`.

1. Vervollständigen Sie die folgenden Seitentabellen, die eine Abbildung des virtuellen Adressraums auf den physikalischen ermöglichen sollen.

Es sind nur Zeilen einzutragen, deren Present-Bit nicht null ist. Die erste Spalte dient zur Angabe der Zeilennummer.

10 Punkte

### Hinweis:

Die Anzahl der Tabellen und Zeilen entspricht nicht der erforderlichen Anzahl.

Fortsetzung von Aufgabe 7.1.

<b>0x3000</b>		<b>0x4000</b>		<b>0x5000</b>	
0x0	0x00 00 40 01		...		
		0x40	0x00 00 10 05		

<b>0x6000</b>		<b>0x7000</b>		<b>0x8000</b>	

2. Wie kann man die vorgestellte MMU so erweitern, dass ein Mischbetrieb von 4 KiB-Seiten und 4 MiB-Seiten (Huges Pages) möglich ist? Geben Sie hierzu auch die nötigen Änderungen an den Tabelleneinträgen der unterschiedlichen Stufen an. 4 Punkte

3. Welche Art von Fragmentierung kann beim Paging auftreten? Wie kann man sie reduzieren? 2 Punkte

4. Welchen Vor- und Nachteil haben mehrstufige Seitentabellen gegenüber einstufigen? 2 Punkte



**Zusätzlicher Platz**

**Zusätzlicher Platz**