

Grundlagen der Rechnerarchitektur und -organisation

.....
 Matrikelnummer Geburtsdatum Vorname Name

- Es sind *keine* Hilfsmittel erlaubt!
- Legen Sie den Ausweis (mit Lichtbild!) griffbereit auf den Platz!
- Dieses Aufgabenheft umfasst 24 Seiten. Überprüfen Sie die Vollständigkeit!
- Gesondert beigelegte Blätter werden nicht bewertet.
- Schreiben Sie deutlich! Unleserliches wird nicht bewertet!
- Es darf nicht mit der Farbe rot geschrieben werden!
- Offensichtlich falsche oder überflüssige Antworten können zu Punktabzug führen!
- Begründen Sie Ihre Antworten!

Durch meine Unterschrift bestätige ich

- den Empfang der vollständigen Klausurunterlagen
- die Kenntnisnahme der obigen Informationen.

Erlangen, den 25.09.2013
 (Unterschrift)

Ich bin damit einverstanden, dass mein Prüfungsergebnis der Klausur unter Angabe der Matrikelnummer veröffentlicht wird.

Erlangen, den 25.09.2013
 (Unterschrift)

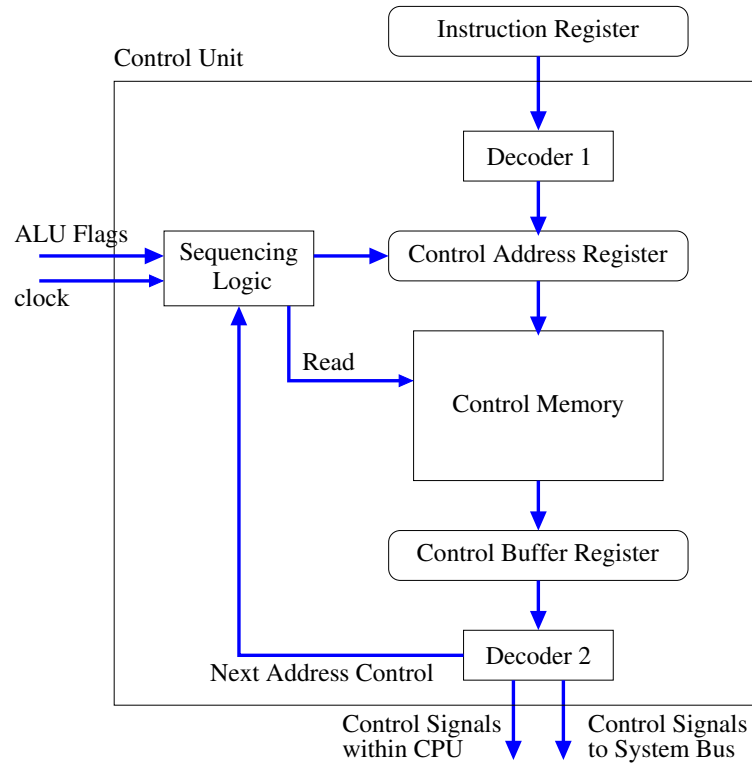
Aufgabe	1	2	3	4	5	6	7
max. Punktzahl	14	17	17	10	11	10	11
erreichte Punktzahl							

Summe	/90				
Bonus	/15				
Gesamt		Note			

Aufgabe 1: Mikroprogrammierung

14 Punkte

Gegeben sei folgende Darstellung der Ablaufsteuerung einer mikroprogrammierten Architektur:



1. Nennen Sie jeweils die Aufgaben der folgenden Einheiten:

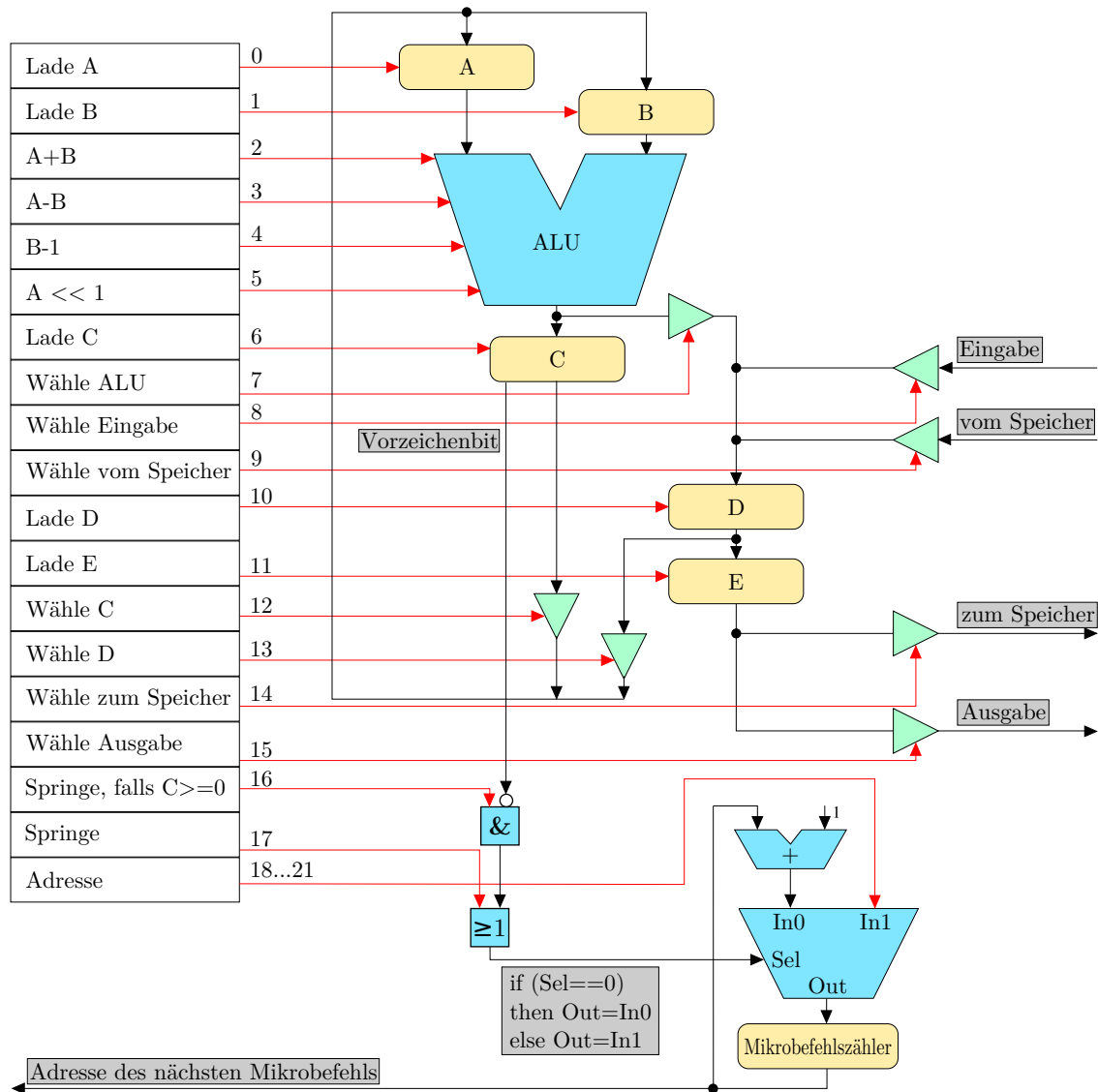
- Sequencing Logic:
- Control Memory:
- Instruction Register:

3 Punkte

2. Welche Form der Mikroprogrammierung wird hier verwendet und woran ist dies erkennbar?

1 Punkt

Gegeben sei der folgende Teil einer CPU:



Eine mikroprogrammierte CPU soll um den Befehl **shl A, width** erweitert werden, der den im **Register A** gespeicherten Wert um **width** Bits nach links verschiebt und das Ergebnis wieder dort ablegt.

Leer gelassene Steuerleitungen entsprechen dem Wert „0“, bei Sprüngen muss die Zieladresse explizit angegeben werden! Das niederwertigste Bit des Sprungziels entspricht Steuerleitung 21.

3. Welche Arten von Sprüngen erlaubt obige Architektur?

Erklären Sie jeweils kurz deren Realisierung!

2 Punkte

4. Implementieren Sie obigen Befehl. Der Operand `width` $\in \mathbb{N} \setminus \{0\}$ kann einmalig vom Speicher gelesen werden. Eine Fehlerbehandlung ist nicht nötig.

Verwenden Sie die folgende Tabelle.

(Die Tabellenlänge entspricht nicht der erwarteten Mikroprogrammlänge!)

8 Punkte

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

Aufgabe 2: Assemblerprogrammierung

17 Punkte

1. Zu welcher Befehlssatzarchitekturklasse zählt die Intel-x86-Architektur? 1 Punkt

2. Welche weiteren Klassen wurden ausführlich behandelt?

Woran kann man jeweils die Klasse anhand ihrer Assemblerinstruktionen erkennen? Geben Sie Beispiele! 6 Punkte

3. Gegeben sei das folgende Intel-x86-Assemblerprogramm:

```

1  .data
2  values:
3      .long 9,4,2,1,6,8,11,3,12,10,5,7
4  length:
5      .long 12
6
7  .text
8  bubsort:
9      pushl %ebx
10     movl $0, %eax
11     jmp .LcondOuter
12  .Louter:
13     movl %eax, %ecx
14     jmp .LcondInner
15  .Linner:
16     movl values(,%eax,4), %edx
17     movl values(,%ecx,4), %ebx
18     cmpl %ebx, %edx
19     jg .Lswap
20     jmp .Lskip
21  .Lswap:
22     movl %ebx, values(,%eax,4)
23     movl %edx, values(,%ecx,4)
24  .Lskip:
25     addl $1, %ecx
26  .LcondInner:
27     cmpl length, %ecx
28     jl .Linner
29     addl $1, %eax
30  .LcondOuter:
31     cmpl length, %eax
32     jl .Louter
33     popl %ebx
34     ret
35
36  .globl main
37  main:
38     call bubsort
39     movl $0, %eax
40     ret

```

Alle Adressen und Daten sind 32 Bit breit. Ein Befehl ist wie folgt aufgebaut (AT&T-Syntax):

Assemblerbefehl	Bedeutung
<code>movl values(,%ecx, 4), %ebx</code>	<code>reg[ebx] = mem[values+reg[ecx]·4]</code>

Wie könnte semantisch äquivalenter Hochsprachencode **ohne goto** aussehen?

Hinweis:

Identifizieren Sie zunächst einzelne Kontrollstrukturen wie Schleifen oder Bedingungen und analysieren Sie anschließend deren Körper.

10 Punkte

Aufgabe 3: Paging

17 Punkte

1. Warum verwendet man beim Paging normalerweise nicht ausschließlich

- sehr große Seiten ($\gg 1$ MiB), bzw.

- sehr kleine Seiten ($\ll 1$ KiB)?

Die gesamte adressierbare Speichergröße soll in beiden Fällen gleich bleiben!

2 Punkte

2. Wozu dient der *Translation Lookaside Buffer*?

1 Punkt

3. Auf einem 64-Bit System soll der Arbeitsspeicher durch mehrstufiges Paging verwaltet werden. Virtuelle und physikalische Adressen seien 48 Bit breit. Die Einträge in den Tabellen umfassen jeweils 8 Byte. Eine Seite sei 4 KiB groß. Die einzelnen Verwaltungstabellen sollen jeweils in einer Page gespeichert werden.

Bestimmen Sie die Tiefe der Tabellenhierarchie rechnerisch!

4 Punkte

4. Eine CPU biete eine Memory-Management-Unit mit folgenden Eigenschaften:

- zweistufige Adresstabellen
- je 1024 Einträge zu je 4 Byte in den Tabellen
 - Bit 31-12: höherwertige Bits der physikalischen Adresse
 - Bit 11-3: unbenutzt
 - Bit 2: Execute-enable-Bit
 - Bit 1: Write-enable-Bit
 - Bit 0: Present-Bit
- Pages zu je 4 KiB Größe

Folgendes soll für ein Programm erfüllt sein:

- im virtuellen Adressbereich `0x00 e0 00 00` bis `0x00 e0 0f ff` soll auf schreibgeschützten Programmcode zugegriffen werden,
- im virtuellen Adressbereich `0x00 03 00 00` bis `0x00 03 ff ff` soll auf Daten zugegriffen werden,
- alle anderen Bereiche sollen Zugriffsfehler auslösen.

Ab der physikalischen Adresse `0x00 c0 00 00` befindet sich ausreichend freier Arbeitsspeicher.

Skizzieren Sie auf der nächsten Seite einen möglichen Aufbau der Page-Tabellen, der obige Bedingungen erfüllt! Tragen Sie jeweils die passenden Adressen in die Tabellen ein, und setzen Sie die Flags entsprechend.

(Bearbeitung auf der nächsten Seite!)

10 Punkte

Fortsetzung von Aufgabe 3.4:

Aufgabe 4: Peripherie

10 Punkte

1. Was sind *Interrupts* und wofür werden sie benötigt? (Stichpunkte!) 2 Punkte

2. Welche drei Formen der Datenübertragung zwischen Ein-/Ausgabegeräten und der CPU, bzw. dem Arbeitsspeicher wurden behandelt?

Geben Sie jeweils ein Beispiel und erklären Sie kurz den Vorteil der entsprechenden Variante.

6 Punkte

3. Was versteht man unter dem *Burst-Modus* bei Datenübertragungen? Was ist der Vorteil?

2 Punkte

Aufgabe 5: Speicherverwaltung

11 Punkte

1. Nennen Sie einen Vorteil und einen Nachteil von Alignment!

2 Punkte

2. Wie viel Platz braucht folgende Datenstruktur auf einer 32-Bit-Architektur, wenn korrektes Alignment verwendet wird? Ändern Sie dabei nicht die Reihenfolge der Variablen!

```
1 | struct student {  
2 |     char [15] name;           //char: 1 Byte  
3 |     char [16] vorname;  
4 |     short   versuch;         //short: 2 Byte  
5 |     int     matnr;           //int: 4 Byte  
6 |     char    geschlecht;  
7 |     float   note;           //float: 4 Byte  
8 | };
```

4 Punkte

3. Wie viele obiger Studenten lassen sich in 96 KiB abspeichern?

1 Punkt

4. Was versteht man unter *Little-Endian* (LE) und *Big-Endian* (BE)?

2 Punkte

5. Ab der Adresse 0x12 stehen folgende Bytes sequentiell im Speicher: 0x76, 0x54, 0x32, 0x10.

Welchem **hexadezimalen** 4-Byte Integer-Wert entspricht dies jeweils in LE- und BE-Darstellung?

2 Punkte

Aufgabe 6: Parallelverarbeitung

10 Punkte

1. Unterscheiden sich Hochsprachenprogramme für VLIW-basierte und superskalare Architekturen?

Wie und von wem wird bei beiden Architekturen die Parallelisierung umgesetzt? 2 Punkte

2. Leiten Sie rechnerisch den asymptotischen Speedup einer Pipeline-Architektur mit s Stufen her! 2 Punkte

3. Warum ist eine sehr hohe Anzahl an Pipelinestufen dennoch nicht sinnvoll?

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

4. Was besagt die Regel von Pollack und wie lässt sich daraus ein Argument für Mehrkernprozessoren ableiten? 2 Punkte

Aufgabe 7: Cache

11 Punkte

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

Wie werden die Lokalitäten jeweils von Caches genutzt?

4 Punkte

2. Eine CPU besitze einen 2048 Byte großen Vollasoziativcache mit einer Blockgröße von 64 Byte. Beim wievielten Speicherzugriff findet frühestens eine Verdrängung aus dem Cache statt? Wie nennt sich die Art des Cache Misses, die dabei auftritt?

3 Punkte

3. Gegeben sei folgende Adressaufteilung:



Welche Organisationsform liegt bei den folgenden Kapazitäten an Nutzdaten vor?

Begründen Sie Ihre Antwort rechnerisch!

a) 512 KiB

b) 2^{16} Byte

4 Punkte

Zusätzlicher Platz

Zusätzlicher Platz