

Klausur zur Vorlesung

Grundlagen der Rechnerarchitektur / Technische Informatik (GRA/TI)

Prof. Marco Platzner
Fachgebiet Technische Informatik
Universität Paderborn

30.09.2011

- Die Bearbeitungsdauer beträgt für alle Studenten **90 Minuten**. Es sind **alle 5 Aufgaben** zu bearbeiten.
- Es sind keine Hilfsmittel zugelassen.
- Schreiben Sie nicht mit Bleistift oder Rotstift.
- Verwenden Sie kein eigenes Papier. Bei Bedarf bekommen Sie Papier bei der Klausuraufsicht.
- Schreiben Sie auf jedes Blatt (auch auf das Konzeptpapier) in Blockschrift Ihren Namen und Ihre Matrikelnummer.
- Bei mehreren präsentierten Lösungen wird die Aufgabe nicht gewertet! Streichen Sie daher bei Angabe mehrerer Lösungsansätze die nicht zu bewertenden Lösungen durch! Verwenden Sie kein Tipp-Ex.
- Abschreiben und abschreiben lassen oder Hilfe Dritter führt zum Nichtbestehen der Klausur.

Nachname: _____

Vorname: _____

Matrikelnummer: _____

Studiengang: _____

Aufkleber

Aufgabe	1	2	3	4	5	Σ
Punkte	15	20	15	20	20	90
Erreicht						

Aufgabe 1 (Multiple Choice)

[15 Punkte]

Bei den folgenden Fragen können keine, eine oder mehrere Antworten richtig sein. Kreuzen Sie die richtigen Antworten deutlich an.

(a) Was sind Aufgaben eines Assemblers?

- ☐ Übersetzen eines Assemblerprogramms in ein Maschinensprachenprogramm.
- ☐ Lesen der ausführbaren Programmdatei von der Festplatte in den Hauptspeicher und Starten des Programms.
- ☐ Verschieben der absoluten Adressen von Instruktionen und Daten.
- ☐ Erzeugen einer Programmdatei im Objektformat.

(b) Was sind Charakteristika einer CISC Instruktionssatzarchitektur?

- ☐ feste Instruktionslänge
- ☐ viele, komplexe Instruktionsarten
- ☐ load/store Architektur
- ☐ viele Varianten von Instruktionen

(c) Was sind Methoden zur Auflösung von Control Hazards?

- ☐ Anhalten der Pipeline
- ☐ Statische Sprungvorhersage
- ☐ Dynamische Sprungvorhersage
- ☐ Forwarding

NAME:

Matrikelnummer:

(d) Was sind Vorteile des dynamischen Pipelinescheduling gegenüber dem statischen Pipelinescheduling:

- ☐ Der Compiler wird vereinfacht.
- ☐ Manche Abhängigkeiten zwischen Instruktionen können nur dynamisch erkannt werden.
- ☐ Der Hardwareaufwand ist geringer.

(e) Die IA-64 (EPIC) Architektur ist eine ...

- ☐ superskalare Architektur
- ☐ Einzyklen-Architektur
- ☐ VLIW Architektur
- ☐ Architektur mit Mehrfachzuordnung

Aufgabe 2 (Assembler)**[20 Punkte]**

Gegeben sind zwei Null-terminierte Zahlenfolgen $\{2,1,4,0\}$ und $\{1,4,0\}$, referenziert durch \$a0 und \$a1 und folgende Assembler-Prozedur lcs:

```
lcs:    addi    $sp,    $sp,    -8
        sw      $ra,    4($sp)
        sw      $s0,    0($sp)

        addi    $v0,    $zero,   $zero
        lw      $t0,    0($a0)
        beq     $t0,    $zero,   return
        lw      $t1,    0($a1)
        beq     $t1,    $zero,   return

        bne     $t0,    $t1,    case2
        addi    $a0,    $a0,    4
        addi    $a1,    $a1,    4
        jal     lcs
        addi    $v0,    $v0,    1
        addi    $a0,    $a0,    -4
        addi    $a1,    $a1,    -4
        j       return

case2:   addi    $a0,    $a0,    4
        jal     lcs
        addi    $s0,    $v0,    $zero
        addi    $a0,    $a0,    -4
        addi    $a1,    $a1,    4
        jal     lcs
        addi    $a1,    $a1,    -4
        bge     $v0,    $s0,    return
        addi    $v0,    $s0,    $zero

return:  lw      $s0,    0($sp)
        lw      $ra,    4($sp)
        addi    $sp,    $sp,    8
        jr      $ra
```

NAME: _____ Matrikelnummer: _____

Matrikelnummer: _____

(a) Skizzieren Sie die Funktion von `lcs`, indem Sie untenstehenden Code ergänzen.

```
int lcs($a0, $a1){
    if ( (memory[$a0] == 0) || (memory[$a1] == 0) )
        return 0;
    if (
```

Ersatzlösung. (Ungültige Lösung streichen.)

```
int lcs($a0, $a1){
    if ( (memory[$a0] == 0) || (memory[$a1] == 0) )
        return 0;
    if (
```

- (b) Betrachten Sie einen Aufruf der Assembler-Prozedur `lcs`. Geben Sie für jede Veränderung des Stackpointers den neuen Wert des Stackpointers, die gesicherten Register und, falls bekannt, die Werte der gesicherten Register an. Gehen Sie davon aus, dass `$v0` mit 0 initialisiert ist. Es ist ausreichend, die ersten zehn Stackveränderungen anzugeben.

	\$sp=156	\$sp=148	\$sp=___	\$sp=___	\$sp=___	\$sp=___	\$sp=___	\$sp=___	\$sp=___	\$sp=___
160	\$ra=?	\$ra=?								
156	\$s0=?	\$s0=?								
152		\$ra=?								
148		\$s0=?								
144										
140										
136										
132										
128										
124										
120										
mem[\$a0]	"2"	"1"								
mem[\$a1]	"1"	"1"								
\$v0	0	0								

Nach den ersten drei Befehlen von `lcs` Nach dem zweiten Aufruf von `lcs`

Ersatzdiagramm: (Ungültige Lösung bitte streichen.)

	\$sp=156	\$sp=148	\$sp=___	\$sp=___	\$sp=___	\$sp=___	\$sp=___	\$sp=___	\$sp=___	\$sp=___
160	\$ra=?	\$ra=?								
156	\$s0=?	\$s0=?								
152		\$ra=?								
148		\$s0=?								
144										
140										
136										
132										
128										
124										
120										
mem[\$a0]	"2"	"1"								
mem[\$a1]	"1"	"1"								
\$v0	0	0								

Nach den ersten drei Befehlen von `lcs` Nach dem zweiten Aufruf von `lcs`

NAME:

Matrikelnummer:

(c) Welchen Rückgabewert berechnet `lcs`?

Aufgabe 3 (Pipelining)

[15 Punkte]

Es sei ein Prozessor mit folgenden Eigenschaften gegeben:

- 5-stufige Pipeline (IF, ID, EX, ME, WB)
- Registerzugriff erfolgt im Halbtaktverfahren
- Getrennter Speicher für Daten und Instruktionen
- Die Pipeline verwendet kein Forwarding

Es sei folgendes Assemblerprogramm gegeben:

```

01:      lw      $5,      100($3)
02:      add     $11,     $4,      $5
03:      mul     $12,     $11,     $5
04:      sub     $13,     $11,     $5
05:      or      $14,     $11,     $5
06:      add     $15,     $11,     $5
07:      sw      $15,     100($3)
  
```

- (a) Lösen Sie die Konflikte indem die Pipeline entsprechend angehalten wird. Erstellen Sie ein Diagramm, aus dem die Pipelinebelegung ersichtlich wird.

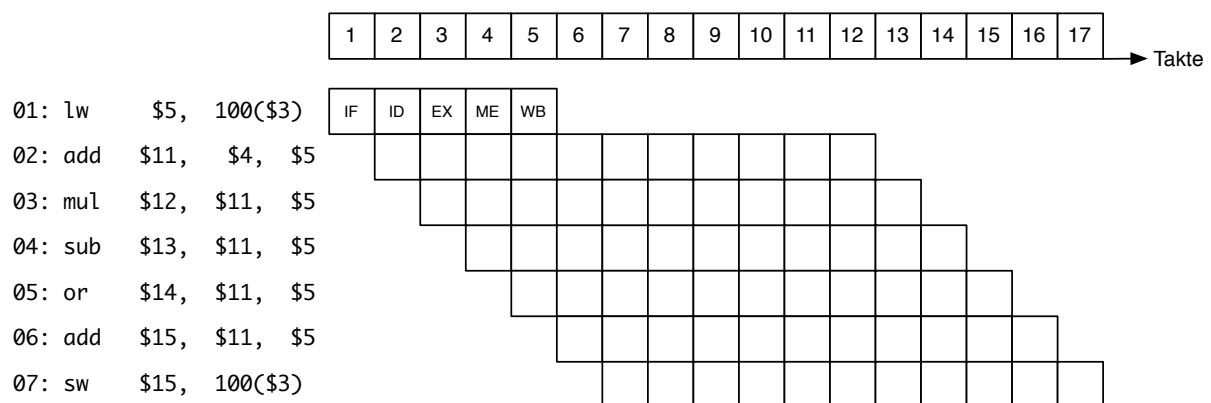


Abbildung 1: Pipelinebelegung

NAME:

Matrikelnummer:

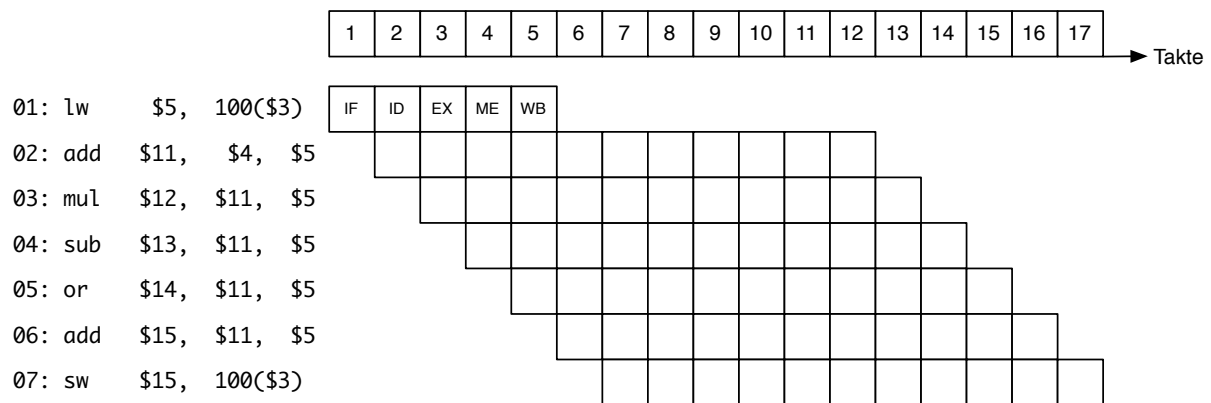


Abbildung 2: (Ersatzdiagramm) Pipelinebelegung

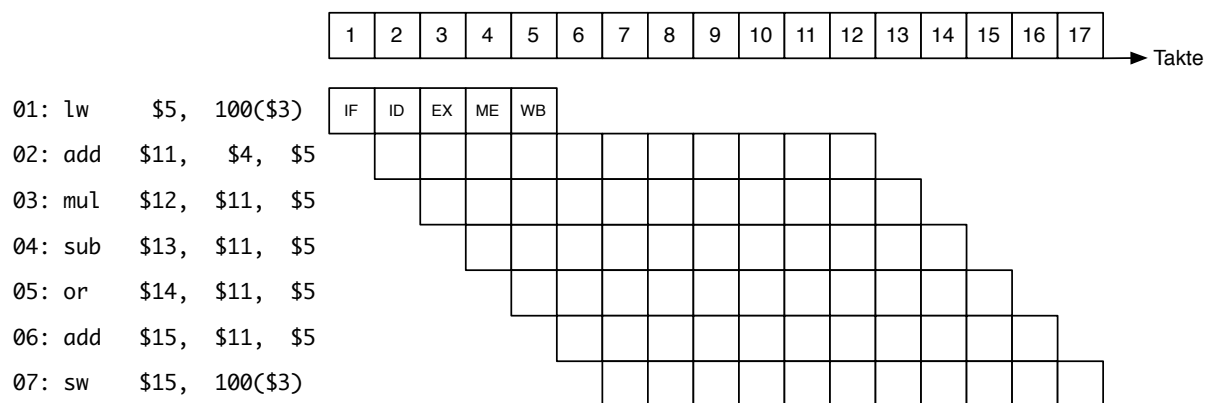


Abbildung 3: (Ersatzdiagramm) Pipelinebelegung

- (b) Gehen Sie nun von einem Prozessor aus der zusätzlich Forwarding unterstützt. Im Fall eines durch das Forwarding nicht auflösbaren Konflikts wird die Pipeline angehalten. Erstellen Sie ein Diagramm, aus dem die Pipelinebelegung für diesen erweiterten Prozessor bei dem gegebenen Assemblerprogramm ersichtlich wird.

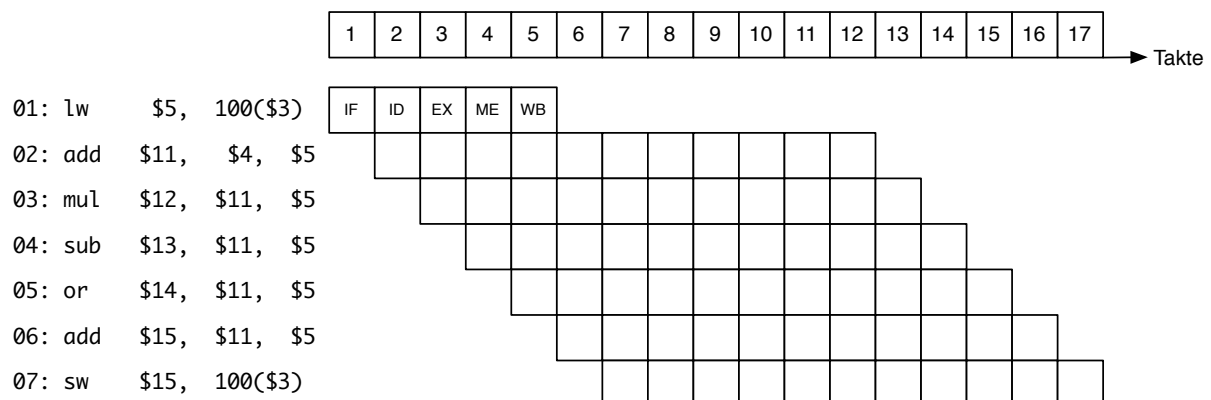


Abbildung 4: Pipelinebelegung

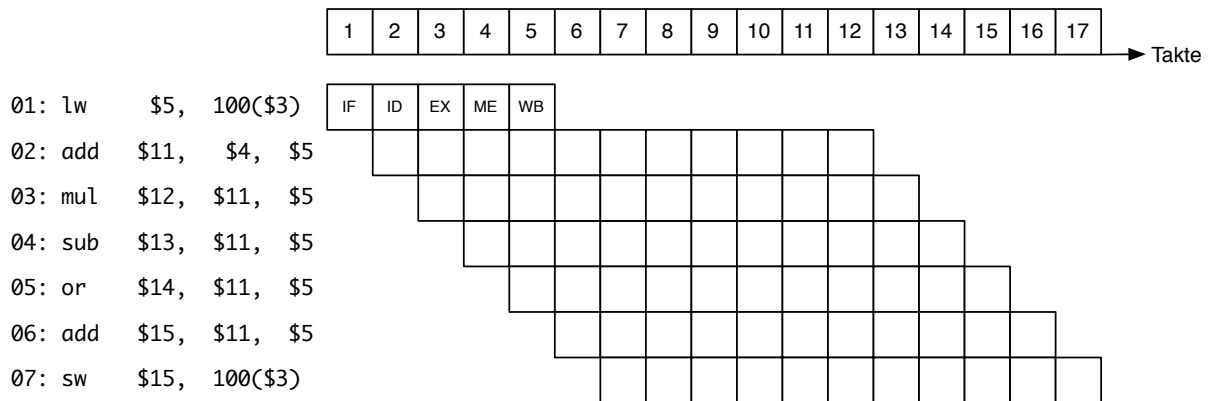


Abbildung 5: (Ersatzdiagramm) Pipelinebelegung

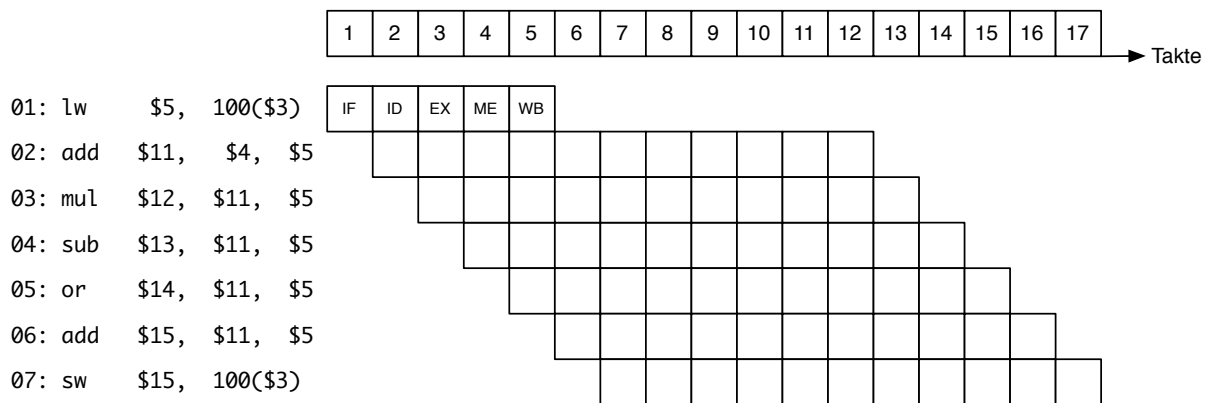


Abbildung 6: (Ersatzdiagramm) Pipelinebelegung

- (c) Berechnen Sie den Speedup zwischen der Lösung mit Forwarding und der Lösung ohne Forwarding. Dokumentieren Sie ihre Berechnung.

NAME:

Matrikelnummer:

Aufgabe 4 (Cacheorganisation)

[20 Punkte]

Index	Satz 1			Satz 2			Satz 3			Satz 4		
	V	Tag	DataByte 3-0	V	Tag	DataByte 3-0	V	Tag	DataByte 3-0	V	Tag	DataByte 3-0
00												
01												
10												
11												

Abbildung 7: Aufbau des Caches

Abbildung 7 zeigt einen Datencache mit einer Nutzdatengröße von **64 Bytes** (1 Wort $\hat{=}$ 8 Bit). Der Datencache ist 4-fach satzassoziativ und hat eine Blockgröße von 4 Bytes. Als Ersetzungsstrategie verwendet der Cache das LRU Verfahren. Der an den Cache angeschlossene Prozessor besitzt eine 8 Bit breite Adressleitung zum Datenspeicher.

- (a) Teilen Sie die 8 Bit Adresse für den gegebenen Cache passend auf, indem Sie in die folgende Abbildung die richtigen Bezeichnungen für jedes Bit einsetzen.

(2 Punkte)

7	6	5	4	3	2	1	0

NAME:

Matrikelnummer:

- (b) Wie groß ist die Gesamtgröße des Datencaches in **Byte**? Geben Sie Ihren Rechenweg mit den Bezeichnungen an!

(3 Punkte)

Gesamtgröße (in Byte) = _____

Index	Satz 1			Satz 2			Satz 3			Satz 4		
	V	Tag	DataByte 3-0	V	Tag	DataByte 3-0	V	Tag	DataByte 3-0	V	Tag	DataByte 3-0
00	1	0000	MEM[0x03-0x00]	0	0001	MEM[0x13-0x10]	0	0010	MEM[0x23-0x20]	0	0011	MEM[0x33-0x30]
01	0	0000	MEM[0x07-0x04]	0	0001	MEM[0x17-0x14]	0	0010	MEM[0x27-0x24]	0	0011	MEM[0x37-0x34]
10	0	0000	MEM[0x0b-0x08]	0	0001	MEM[0x1b-0x18]	0	0010	MEM[0x2b-0x28]	0	0011	MEM[0x3b-0x38]
11	0	0000	MEM[0x0f-0x0c]	0	0001	MEM[0x1f-0x1c]	0	0010	MEM[0x2f-0x2c]	0	0011	MEM[0x3f-0x3c]

Abbildung 8: Cachebelegung vor Zugriffssequenz

- (c) Gegeben ist die in Abbildung 8 dargestellte Cachebelegung. Ein Programm führt nun folgende Sequenz an Lesezugriffen auf dem Datenspeicher aus (Adressen in Hexadezimal):

0x01, 0x0d, 0x2a, 0x88, 0x59, 0x77, 0xfa, 0xea, 0x75, 0x28, 0x4d, 0x5b

Füllen Sie nach jedem Zugriff die angegebene Zeile aus. Unter *H/M* tragen Sie ein **H** (für Hit) ein, falls die Daten aus dem Cache geladen werden können, andernfalls ein **M** (für Miss). Tragen Sie weiterhin den verwendeten *Satz*, den entsprechenden *Index* und *Tag* sowie das *Valid Bit (V)* und die *Daten* nach dem Lesezugriff ein.

(15 Punkte)

			Satz: ____
	H/M	Index	V Tag DataByte 3-0
0x01:	___	___	MEM[___ - ___]
			Satz: ____
	H/M	Index	V Tag DataByte 3-0
0x0d:	___	___	MEM[___ - ___]
			Satz: ____
	H/M	Index	V Tag DataByte 3-0
0x2a:	___	___	MEM[___ - ___]
			Satz: ____
	H/M	Index	V Tag DataByte 3-0
0x88:	___	___	MEM[___ - ___]
			Satz: ____
	H/M	Index	V Tag DataByte 3-0
0x59:	___	___	MEM[___ - ___]

NAME:

Matrikelnummer:

		Satz: _____
H/M	Index	V Tag DataByte 3-0
0x77:	_____	MEM[_____ - _____]

		Satz: _____
H/M	Index	V Tag DataByte 3-0
0xfa:	_____	MEM[_____ - _____]

		Satz: _____
H/M	Index	V Tag DataByte 3-0
0xea:	_____	MEM[_____ - _____]

		Satz: _____
H/M	Index	V Tag DataByte 3-0
0x75:	_____	MEM[_____ - _____]

		Satz: _____
H/M	Index	V Tag DataByte 3-0
0x28:	_____	MEM[_____ - _____]

		Satz: _____
H/M	Index	V Tag DataByte 3-0
0x4d:	_____	MEM[_____ - _____]

		Satz: _____
H/M	Index	V Tag DataByte 3-0
0x5b:	_____	MEM[_____ - _____]

Aufgabe 5 (IO Performance, Audio Stream Server)

[20 Punkte]

Es soll ein Server für Audio-Streams konzipiert werden (siehe Abbildung 9). Für die Hardware-Konfiguration des Servers gelten folgende Bedingungen:

- Die CPU hat eine Datenwortbreite von 64 Bit und erreicht 50.000 MIPS.
- Der Speicherbus läuft mit 100 MHz und transportiert 4 Worte pro Bus-Takt.
- Neben der CPU können mehrere Disk-Controller an den Speicherbus angeschlossen werden.
- An einen Disk-Controller können bis zu 8 Disks mit einer Transferrate von jeweils $6 \cdot 10^9$ Byte pro Sekunde angeschlossen werden.
- Die verwendeten Disks haben eine Transferrate von $1024 \cdot 10^6$ Bit pro Sekunde. Die *average seek time* beträgt 4 ms, die *average rotational latency* beträgt 3 ms.
- In einer I/O-Operation werden $128 \cdot 10^3$ Byte an Daten transferiert. Dazu benötigt das Anwendungsprogramm 600.000 Instruktionen und das Betriebssystem 400.000 Instruktionen.

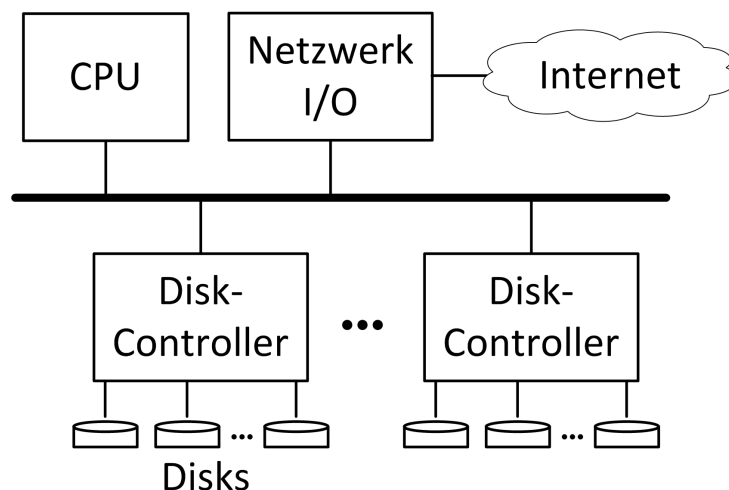


Abbildung 9: Architekturdiagramm des Audio-Stream Servers

Geben Sie zu den einzelnen Teilaufgaben unbedingt den jeweiligen Rechenweg an!

NAME:

Matrikelnummer:

- (a) Wie hoch ist die Transferrate des Speicherbusses?

- (b) Wie viele I/O-Operationen pro Sekunde (IOPS) kann die CPU durchführen?

- (c) Wie viele IOPS können über den Speicherbus transferiert werden?

- (d) Wie lange dauert ein Disk-Zugriff für eine I/O-Operation durchschnittlich? Nehmen Sie für Ihre Berechnung an, dass alle für eine I/O-Operation benötigten Datenblöcke in sequentiellen Sektoren der Disk abgelegt sind.

- (e) Wie viele I/O-Operationen pro Sekunde können von einer Disk bedient werden?

- (f) Das Verteilen von Audio-Streams mit einer Datenrate von $128 \cdot 10^3$ Bit pro Sekunde ist auf dem Server als Sequenz von I/O-Operationen realisiert. Wie viele Disks und wie viele Disk-Controller werden benötigt, um 20.000 Nutzer zeitgleich mit individuellen Audio-Streams zu versorgen? Nehmen Sie für Ihre Berechnung an, dass es zu keinen Blockierungen bei den Disk Zugriffen kommt und dass die Daten von den Disk-Controllern direkt zur Netzwerkschnittstelle transferiert werden können.
- (g) Um Energie zu sparen soll die Taktfrequenz des Prozessors verringert werden. Um welchen Faktor kann man den CPU Takt höchstens drosseln, um weiterhin 20.000 Nutzer zeitgleich mit Audio-Streams versorgen zu können?

NAME:

Matrikelnummer:

Konzeptpapier: Falls der Platz unter den einzelnen Aufgaben nicht ausreicht, können Sie diese Seiten für Zwischenrechnungen nutzen. Bitte Lösung und Lösungsweg eindeutig mit der Aufgabennummer markieren!

Konzeptpapier: Falls der Platz unter den einzelnen Aufgaben nicht ausreicht, können Sie diese Seiten für Zwischenrechnungen nutzen. Bitte Lösung und Lösungsweg eindeutig mit der Aufgabennummer markieren!