

Aufgabe 1: Instruktionssatzarchitektur (14 Punkte)

- Vier Klassen von Befehlssatz-Architekturen wurden in der Vorlesung vorgestellt. Zu welcher Klasse gehört eine CPU, die nachfolgenden Assembler-Code ausführen kann? Begründen Sie Ihre Antwort!
(2 Punkte)

```
load a, %r0
load b, %r1
add %r0, %r1, %r2
sub %r0, %r1, %r0
mul %r2, %r0, %r0
store %r0, c
```

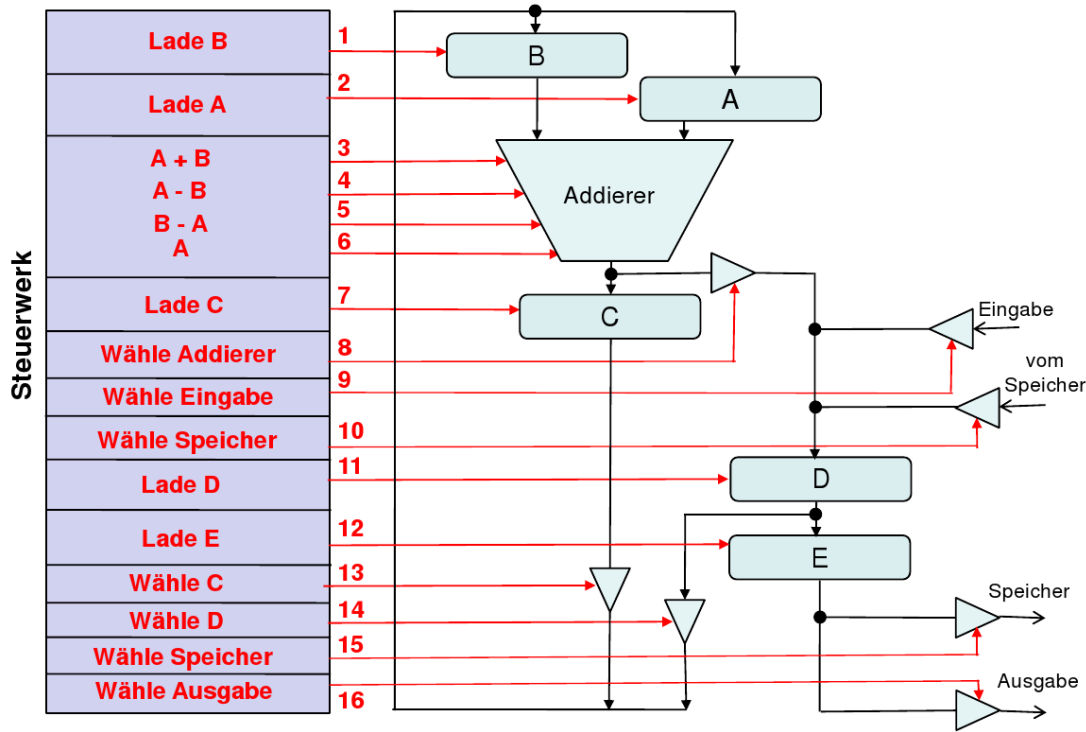
- Schreiben Sie obigen Assembler-Code um, sodass er für eine CPU der Klasse „Akkumulator“ geeignet ist! Sie dürfen die Speicherstellen `tmp0`, `tmp1` usw. für Zwischenergebnisse nutzen.
(2 Punkte)

- Moderne CPUs nutzen Pipelining mit vielen Pipeline-Stufen. Wie groß ist der Speed-Up einer Pipeline mit N Stufen? Begründen Sie die Formel für den Speed-Up!
(3 Punkte)
- Welche Bedingungen müssen erfüllt sein, damit beim Pipelining mit jedem Takt die Bearbeitung einer Instruktion abgeschlossen werden kann?
(3 Punkte)
- Bei der Abarbeitung von Sprüngen sind bei CPUs mit Pipelining Besonderheiten zu beachten. Erklären Sie diese!
(4 Punkte)

- Schreiben Sie Assembler-Code für obiges Hochsprachen-Programm!
(10 Punkte)

Aufgabe 3: Mikroprogrammierung (12 Punkte)

Gegeben sei der aus der Vorlesung/Übung bekannte Teil einer CPU:



- Schreiben Sie ein Mikroprogramm, das nacheinander die Werte 0, 1, 2, ..., 7 auf der Ausgabe ausgibt! Die Register *B*, *C*, *D* und *E* enthalten bei Programmstart jeweils den Wert 0. Der Wert von *A* darf gewählt werden. (8 Punkte)

- Wie müsste die Hardware-Struktur verändert werden, um die 8 Zahlen mit weniger Takten ausgeben zu können? Begründen Sie Ihre Antwort!
(4 Punkte)

Aufgabe 4: I/O (9 Punkte)

Gegeben sei folgendes Programm:

```
int array[10];
...
read_values_from_disk(array);
sum = calculate_sum(array);
...
```

Zur Auswahl stehen 6 Rechner:

Rechner A: ohne DMA, ohne Cache

Rechner B: mit DMA, ohne Cache

Rechner C: ohne DMA, mit großem Cache (Write-Through)

Rechner D: mit DMA, mit großem Cache (Write-Through)

Rechner E: ohne DMA, mit großem Cache (Write-Back)

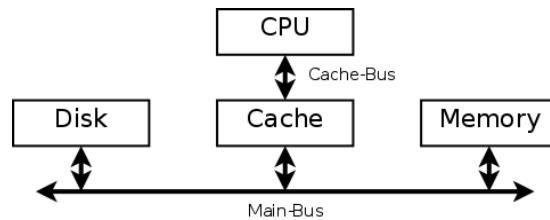
Rechner F: mit DMA, mit großem Cache (Write-Back)

Abgesehen von diesen Unterschieden sind die Rechner ansonsten gleich.

- Was bedeutet „Direct Memory Access“?
(1 Punkt)

- Was bedeuten „Write-Through“ und „Write-Back“?
(2 Punkte)

- Über welche Busse und durch welche Komponenten werden die Daten bei den 6 Rechnern jeweils geschleust?
(3 Punkte)



- Welcher dieser 6 Rechner wird das Programm am schnellsten abgearbeitet haben (keine nebenläufigen Prozesse)? Begründen Sie Ihre Antwort!
(3 Punkte)

Aufgabe 5: Speicher/Speicherhierarchie (15 Punkte)

- Warum wird der Speicher aus Hintergrundspeicher, Hauptspeicher und Caches hierarchisch aufgebaut?
(1 Punkte)

- Der Hauptspeicher wird häufig mit sogenanntem „Interleaving“ betrieben. Was bedeutet dies? Warum macht man das so?
(4 Punkte)

- Ein Rechner habe einen Bus von der CPU zum Hauptspeicher mit 32-Bit Breite sowie 4-fach-Interleaving. Auf welche Hauptspeicher-Bank greift die CPU zu, wenn sie ein Byte von der physikalischen Adresse 0x12345678 liest? Begründen Sie Ihre Antwort!
(2 Punkte)

- Was ändert sich, wenn die CPU statt von der physikalischen Adresse 0x12345678 von der virtuellen Adresse 0x12345678 liest? Begründen Sie Ihre Antwort!
(1 Punkt)

- Was wird als „räumliche Lokalität“ und was als „zeitliche Lokalität“ bezeichnet?
(2 Punkte)

- Die Matrix-Multiplikation ist definiert als

$$\begin{pmatrix} a_{1,1} & a_{2,1} & \cdots & a_{N,1} \\ a_{1,2} & a_{2,2} & \cdots & a_{N,2} \\ \cdots & \cdots & & \cdots \\ a_{1,M} & a_{2,M} & \cdots & a_{N,M} \end{pmatrix} * \begin{pmatrix} b_{1,1} & b_{2,1} & \cdots & b_{L,1} \\ b_{1,2} & b_{2,2} & \cdots & b_{L,2} \\ \cdots & \cdots & & \cdots \\ b_{1,N} & b_{2,N} & \cdots & b_{L,N} \end{pmatrix} = \begin{pmatrix} c_{1,1} & c_{2,1} & \cdots & c_{L,1} \\ c_{1,2} & c_{2,2} & \cdots & c_{L,2} \\ \cdots & \cdots & & \cdots \\ c_{1,M} & c_{2,M} & \cdots & c_{L,M} \end{pmatrix}$$

mit

$$c_{i,j} = \sum_{k=1}^N a_{k,j} b_{i,k}$$

Wenn L , M und N sehr große Zahlen sind, ergeben sich bei der Berechnung der $c_{i,j}$ Probleme beim Caching, wenn man versucht, die Summen nach obiger Formel direkt zu berechnen. Warum?

(5 Punkte)

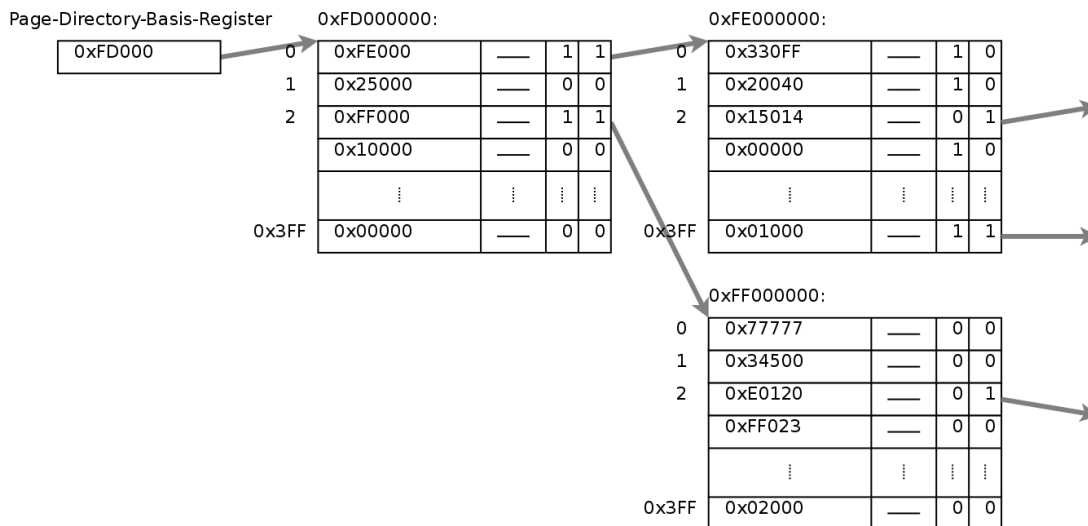
Aufgabe 6: Segmentierung/MMU (18 Punkte)

- Segmente sind definiert durch ihre Basis-Adresse und ihre Länge. Wie verwendet die CPU diese beiden Werte beim Zugriff auf ein Segment?
(4 Punkt)
- Wo werden Segment-Basis-Adressen und -Längen i.A. gespeichert?
(1 Punkt)
- Je größer der Adressraum einer CPU ist, desto mehr Stufen haben i.A. die Page-Tabellen. Warum?
(2 Punkte)

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 11-2: unbenutzt
 - Bit 1: Write-Protect-Bit
 - Bit 0: Present-Bit
- Pages zu je 4 KByte Größe

Gegeben seien folgende Page-Tables:



- Welche Einträge stehen in einem voll-assoziativen TLB mit vier Einträgen, nachdem die CPU nacheinander auf die virtuellen Adressen 0x003FF888, 0x003FF100, 0x00802100 und 0x00002400 lesend zugegriffen hat? (10 Punkte)

- Was passiert, wenn die CPU auf die virtuelle Adresse 0x003FF888 schreibend zugreift? Begründen Sie Ihre Antwort!
(1 Punkt)

Aufgabe 7: Multi-Threading/Multi-Core (17 Punkte)

- Was ist der Unterschied von „Multi-Threading“ und „Multi-Core“?
(2 Punkte)

- Gegeben seien eine 4-Core-CPU, eine CPU mit 4-fach-Threading und eine CPU mit nur einem Core aber vierfacher Taktfrequenz. All diese CPUs können die gleiche maximale Anzahl von Instruktionen pro Zeiteinheit abarbeiten.

Beschreiben Sie die Vor- und Nachteile der CPUs! Geben Sie je einen Einsatzzweck an, für die die verschiedenen CPUs besonders geeignet sind!

(9 Punkte)

- Eine Multi-Core-CPU wird mit einer Speicherhierarchie verbunden, die aus Hintergrundspeicher, Hauptspeicher, 3rd-Level-, 2nd-Level- und 1st-Level-Caches besteht. Nur Teile der Speicherhierarchie werden von allen Cores gemeinsam genutzt.
 - Geben Sie ein sinnvolles Beispiel an, welche Speicher gemeinsam und welche nicht gemeinsam genutzt werden! (2 Punkte)

 - Warum werden nicht alle Bereiche gemeinsam genutzt? Warum hat nicht jeder Core seine eigene Speicherhierarchie? (2 Punkte)

- Eine 4-Core-CPU kann eine MMU oder auch 4 MMUs besitzen. Geben sie Vor- und Nachteile an! (2 Punkte)

Zusätzlicher Platz

Zusätzlicher Platz

Zusätzlicher Platz