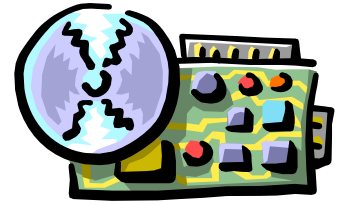




TI II:  
Rechnerarchitektur  
SS 2007  
Übungsblatt Nr. 5



Prof. Dr.-Ing. Jochen Schiller, AG Technische Informatik, Freie Universität Berlin

**Ausgabe am 18.06.2007 — Abgabe spätestens 06.07.2007, 12:00 Uhr**

Bitte bei der Abgabe beide Namen/Matr.Nr. der Mitglieder einer Gruppe, NUMMER DER ÜBUNG/TEILAUFGABE und DATUM auf den Lösungsblättern **nicht vergessen!** Darauf achten, dass die Lösungen beim richtigen Tutor/der richtigen Tutorin abgegeben werden.

**Zu spät abgegebene Lösungen werden nicht mehr berücksichtigt!**

---

## 1. Aufgabe: Virtueller Speicher (9 Punkte)

Ein Computer habe einen virtuellen Adressraum mit 128 Seiten, aber lediglich 4 Seitenrahmen. Anfangs ist der Speicher bereits mit den Seiten 9, 4, 1, 5 (in dieser Reihenfolge) gefüllt worden.

Ein Programm referenziert die virtuellen Seiten in folgender Reihenfolge:

3, 1, 2, 4, 2, 4, 7, 6, 3, 4, 1, 9

- Welche Referenzen verursachen einen Seitenfehler bei LRU als Ersetzungsstrategie? (2 Punkte)
- Welche Referenzen verursachen einen Seitenfehler bei LIFO als Ersetzungsstrategie? (2 Punkte)
- Welche Referenzen verursachen einen Seitenfehler bei FIFO als Ersetzungsstrategie? (2 Punkte)
- Nennen/Beschreiben Sie jeweils eine Zugriffsfolge, die in den obigen Verfahren zu einem Maximum an Seitenfehlern führt. (3 Punkte)

## 2. Caching (10 Punkte)

- Welchen Vorteil bringt die Aufteilung in Tags und Index zur Adressierung eines Datums im Cache? (1 Punkt)
- An wie vielen Stellen muss ein Block bei voll-assoziativ, direct-mapped und n-way-set Caches gesucht werden? (3 Punkte)
- Was versteht man unter Cache-Kohärenz? Warum strebt man nicht Konsistenz an? (2 Punkte)
- Welche generellen Vor- und Nachteile weisen virtuell bzw. physikalisch adressierte Cache-Speicher auf? (2 Punkte)
- Wie wirken sich Prozesswechsel auf den jeweiligen Cache aus? Beachten Sie die Position der MMU! (2 Punkte)

## 3. Aufgabe: Größe der Tags (4 Punkte)

Eine erhöhte Assoziativität benötigt mehr Vergleiche und mehr Tag-Bits pro Cache-Block. Ermitteln Sie für einen Cache mit 4k-Blöcken, einer Blockgröße von 4 Wörtern und einer 32-Bit-Adresse die Gesamtzahl der Sätze sowie die Gesamtzahl der Tag-Bits für direkt abgebildete, zweifach und vierfach satzassoziative und vollassoziative Cache-Organisationen.

## 4. Aufgabe: Cache-Zugriff (6 Punkte)

Gegeben sei ein vollassoziativer Datencache, Größe 1024 KByte, am Anfang leer, mit 64 Byte pro Cache-Zeile. Es werde folgende Programmschleife zur Addition von 128 16-Bit-Zahlen, die in einem Array `g[]` liegen, durchgeführt:

```
int i=0;
for(j=0; j<128; j++)
    i = i+g[j];
```

Weitere Annahmen: Bei einem Cache-Miss wird immer die ganze Cache-Zeile in den Cache geladen. Die Daten des Arrays `g[]` liegen im Speicher an direkt aufeinander folgenden Adressen und in der Reihenfolge, in der sie auch von der Schleife eingelesen werden.

1. Tragen Sie die Gesamtzahl der Cache-Misses und Cache-Hits je Zeile in die Abbildung 1 ein.
2. Berechnen Sie die Cache-Miss-Quote.

		Miss	Hit
	write i = 0		
	write j = 0		
loop:	read j		
	if (j >= 128) exit else		
	read g[j]		
	read i		
	compute i + g[j]		
	write i		
	read j		
	compute j + 1		
	write j		
	jump to loop		

## 5. Aufgabe (7 Punkte)

In einem Mikroprozessorsystem 32-bit-Datenzugriff auf den Hauptspeicher ist ein Cache vorhanden. Eine Cacheline besteht aus je acht Bytes. Die Hauptspeicheradresse umfasst 24 Bits und die Kapazität des Cache beträgt 256 Bytes.

- a) Wie viele Cache-Zeilen gibt es? (1 Punkt)
- b) Skizzieren Sie die Unterteilung der Hauptspeicheradresse für direct-mapped-Cache, vollassoziativen Cache und 4-way-set-assoziativ-Cache. (3 Punkte)
- c) Wie viele Vergleiche werden für jeden Cache benötigt und welche Bitbreite haben sie? (3 Punkte)

	Vergleicher	Tag-Länge
direct-mapped		
vollassoziativ		
4-way-set		

## 6. Aufgabe (8 Punkte)

In dieser Aufgabe sollen verschiedene Cache-Organisationsformen am Beispiel eines einfachen Verschlüsselungsalgorithmus untersucht werden. Gegeben sei ein 4-elementiger Vektor  $k$ , der den Schlüssel enthält. Der Algorithmus arbeitet wie folgt, wobei  $\otimes$  die XOR-Verknüpfung repräsentiert:

```
FOR i= 0 TO 15 DO
    c[i]=d[i]  $\otimes$  k[i mod 4]
END
```

Die zu verschlüsselnden Daten werden dem Algorithmus in Blöcken zu je 16 Worten im Vektor  $d$  übergeben. Das Ergebnis wird im Vektor  $c$  gespeichert. Dieser Algorithmus soll auf einem wortadressierten 16-Bit-Prozessor ausgeführt werden.

- Der Schlüssel  $k$  ist unter der Adresse  $0 \times 1E30$  abgelegt.
- Der Vektor  $d$  beginnt bei Adresse  $0 \times 4AC0$ .
- $c$  beginnt bei Adresse  $0 \times 4BC0$ .

Gegeben seien ein direkt-abgebildeter Cache (DM) und ein 4-fach satzassoziativer Cache (A4). Beide Caches haben eine Kapazität von 256 Worten, wobei eine Cache-Line jeweils 4 Worte umfasst. Die Caches seien zu Beginn der Ausführung des Algorithmus leer.

- a) Bestimmen Sie für beide Caches die Anzahl der Sätze sowie die Anzahl der Cache-Lines pro Satz. (2 Punkte)
- b) Geben Sie für DM die Nummern aller Sätze an, die durch den Algorithmus in Anspruch genommen werden. (2 Punkte)
- c) Ermitteln Sie die Trefferrate für DM. (1 Punkt)
- d) Geben Sie für A4 die Nummern aller Sätze an, die durch den Algorithmus in Anspruch genommen werden. (2 Punkte)
- e) Ermitteln Sie die Trefferrate für A4. (1 Punkt)