



Prof. Dr. Rolf Drechsler, drechsler@informatik.uni-bremen.de, MZH 3510  
Dr. Nicole Drechsler nd@informatik.uni-bremen.de, MZH 3550

4. Übungsblatt zur Vorlesung

# Technische Informatik 1

## Aufgabe 1

(3 Punkte)

In der Vorlesung haben wir zwei Organisationsformen für Caches kennengelernt: den *vollassoziativen Cache* (VA) und den *direct-mapped Cache* (DM).

- Stelle in Tabellenform die Vor- und Nachteile beider Verfahren einander gegenüber. Erläutere außerdem jeden aufgeführten Tabelleneintrag mit einer entsprechenden Begründung.
- Überlege Dir dann, wie man die Nachteile beider Verfahren durch eine *Kombination* beider Organisationsformen überwinden könnte (Hinweis: eine solche Kombination gibt es wirklich, diese ist unter dem Namen *n-fach assoziativer Cache* bekannt).

## Aufgabe 2

(3 Punkte)

Beim Entwurf eines Rechners wird als Cache Speicher ein *direct-mapped cache* vorgesehen.

- Beim entgeltigen Entwurf werden versehentlich statt der  $i$  niederwertigsten die  $i$  höchstwertigen Bits in den Dekoder geschoben, während die restlichen Bits als Adress-TAG verwendet werden. Bei nachfolgenden Messungen stellt man fest, dass die Trefferrate nicht wie erwartet bei 96.7 % sondern deutlich unter 60 % liegt. Wie ist dies zu erklären?
- Vor der Entscheidung zu Gunsten eines *direct-mapped Caches* wurde eine neue Schaltung für die Realisierung der *LRU* Verdrängungsstrategie entwickelt, durch die gegenüber herkömmlichen Schaltungen 10 % der Gatter eingespart werden können. Warum ist dies im vorliegenden Fall bedeutungslos?

## Aufgabe 3

(7 Punkte)

In einer CPU werde ein *vollassoziativer Cache* mit 4 Cacheblöcken verwendet.

- Innerhalb eines laufenden Programms auf der oben beschriebenen CPU wird auf die folgenden Adressen nacheinander zugegriffen.

$0x1A2$ ,  $0x1A4$ ,  $0x1A6$ ,  $0x1A8$ ,  $0x1A0$ ,  $0x1A2$ ,  $0x1A6$ ,  $0x1C0$ ,  
 $0x1A6$ ,  $0x1A8$ ,  $0x1A6$ ,  $0x1B4$ ,  $0x1C0$ ,  $0x1C2$ ,  $0x1C0$ ,  $0x1A6$

(Anmerkung: Das Prefix  $0x$  kennzeichnet einen Wert in Hexadezimaldarstellung, also zur Basis 16:

$$0xA1D = 10 \cdot 16^2 + 1 \cdot 16^1 + 13 \cdot 16^0 = 2589)$$

Aus den Datenwörtern welcher Adressen setzen sich die Blöcke des Cache nach den jeweiligen Zugriffen zusammen, wenn die *first-in-first-out* (FIFO)-Strategie benutzt wird? Zu wievielen Cache-Misses kommt es?

- Wie ist dagegen das Verhalten wenn die *least-recently-used* (LRU)-Verdrängungsstrategie verwendet wird?
- Verhält sich die LRU-Strategie immer besser als die FIFO-Strategie? Gib ein Beispiel zur Erklärung an.
- Bei welcher der beiden Verdrängungsstrategien kann die Entscheidung, welcher Cache-Block ersetzt wird, schneller getroffen werden? Begründe deine Antwort!

**Aufgabe 4**

(3 Punkte)

Beim Entwurf eines Rechners ist man sich unklar darüber, ob man ihn mit einem *direct-mapped Cache* (DM), einem *2-fach assoziativen Cache* (2A) oder einem *vollassoziativen Cache* (VA) ausstatten soll.

Die geschätzten technischen Daten für die zur Verfügung stehenden Caches:

Cache	Trefferrate	Zugriffszeit des Cache	Zugriffszeit zum Hauptspeicher
DM	97.1 %	23.9ns	196ns
2A	96.2 %	21.6ns	200ns
VA	96.9 %	22.1ns	198ns

(Dabei gibt die Trefferrate an, bei wieviel Prozent der Speicherzugriffe ein Cache-Hit vorkommt.) Für welchen Cache sollte man sich entscheiden, wenn keine weiteren Kriterien berücksichtigt werden müssen?

**Aufgabe 5**

(4 Punkte)

Gegeben sei ein Rechner mit virtueller Speicherverwaltung durch Paging. Der Hauptspeicher enthält vier Seitenrahmen. In einem Testlauf wird nacheinander auf die Seiten

0x100, 0x0FF, 0x113, 0x10A, 0x113, 0x10B,  
0x10B, 0x100, 0x0FF, 0x10C, 0x114, 0x100

zugegriffen. Wie sieht die Seitentabelle nach den einzelnen Zugriffen aus, wenn als Verdrängungsstrategie

- a) Least-Recently-Used (LRU)
  - b) Least-Frequently-Used (LFU)
- verwendet wird?

**Abgabetermin: zu Beginn der Vorlesung am 22.05.2008**