

Prof. L. Thiele

## Technische Informatik 1 - HS 2011

---

### Lösungsvorschläge für Übung 10

Datum: 15.12.2011

---

## 1 Speicherhierarchien (Cache)

Gegeben seien drei Rechner mit unterschiedlichen Cache-Konfigurationen:

- Rechner A: direkte Abbildung (direct-mapped).
- Rechner B: teilassoziativ mit 2 Einträgen pro Index (2-way set associative).
- Rechner C: vollassoziativ.

Jeder Cache habe eine Kapazität von 4 Blöcken á 2 Worte.

- (a) Berechnen Sie für jeden Rechner die effektive Cachegrösse in Bit. Gehen Sie hierbei von einer 32-Bit-Byteadressierung aus. Von den Statusbits soll lediglich das Valid-Bit berücksichtigt werden, andere Statusbits (Dirty-Bit, LRU, etc.) sind in dieser Teilaufgabe nicht zu berücksichtigen.

### Lösung:

Für jeden Block werden 1 Bit für das Valid-Feld,  $2 \times 32$  Bit für die Daten und eine gewisse Anzahl von Bits für das Tag-Feld benötigt.

Die maximale Tagbreite von 32 Bit reduziert sich je nach Cachearchitektur. Die benötigte Tagbreite reduziert sich bei allen drei Architekturen um 3 Bit, da die Maschine Byte-Adressen verwendet aber nur Worte adressierbar sind (byte offset: 2 Bit) und weil jede Cacheline 2 Worte speichert (block offset: 1 Bit).

Zudem reduziert sich die Breite des Tag-Felds mit der Anzahl der Indizes, die der Cache besitzt. Da Rechner A vier Indizes besitzt, reduziert sich die Breite um 2 Bit und bei Rechner B (zwei Indizes) um 1 Bit. Rechner C hat nur einen Index und muss deshalb alle 29 verbleibenden Bits speichern.

- Rechner A: Tag-Grösse =  $(32-3-2)$  Bit = 27 Bit.
- Rechner B: Tag-Grösse =  $(32-3-1)$  Bit = 28 Bit.
- Rechner C: Tag-Grösse =  $(32-3-0)$  Bit = 29 Bit.

Die Cachegrösse berechnet sich nun durch Multiplikation von Blockanzahl mit effektiver Blockgrösse:

- Rechner A:  $4 \times (1+64+27)$  Bit = 368 Bit.
- Rechner B:  $4 \times (1+64+28)$  Bit = 372 Bit.
- Rechner C:  $4 \times (1+64+29)$  Bit = 376 Bit.

- (b) Nach dem Einschalten des Rechners wird sequentiell auf die folgenden *Wortadressen* zugegriffen: 19, 10, 19, 14, 11, 15. Wie ist der Zustand der einzelnen Caches vor und nach jedem Speicherzugriff? Wann treten Fehlzugriffe (Misses) auf? Nennen Sie für jeden Cache die Zahl der insgesamt auftretenden Fehlzugriffe.

Nehmen Sie an, dass als Blockersetzungsschema das LRU-Verfahren (least recently used) benutzt wird, d.h. der am längsten unbenutzte Block wird ersetzt.

**Lösung:**

Die folgenden Tabellen zeigen jeweils den Zustand des Caches nach dem Zugriff auf die Adresse, welche in der ersten Spalte gegeben ist. Zusätzlich ist in der Tabelle auch die binäre Darstellung der Adresse und ihre Zerlegung in die Anteile Tag (tag), Index (idx), Word offset (owr) und Byte offset (oby) dargestellt. Der Byteoffset ist für alle Adresse identisch (00), da die Adressen in der Aufgabe als *Wortadressen* gegeben sind.

- Rechner A:

Anzahl Misses: 5

Adr	Adr binär				Hit/Miss	Ind 00	Ind 01	Ind 10	Ind 11
	tag	idx	owr	oby					
19	10	01	1	00	Miss compul		18,19		
10	01	01	0	00	Miss confl		10,11		
19	10	01	1	00	Miss confl		18,19		
14	01	11	0	00	Miss compul		18,19		14,15
11	01	01	1	00	Miss confl		10,11		14,15
15	01	11	1	00	Hit		10,11		14,15

- Rechner B:

Anzahl Misses: 4

Adr	Adr binär $\frac{1}{2}r$				Hit/Miss	Ind 0		Ind 1	
	tag	idx	owr	oby		Block 0	Block 1	Block 0	Block 1
19	100	1	1	00	Miss comput			18,19	
10	010	1	0	00	Miss comput			18,19	10,11
19	100	1	1	00	Hit			18,19	10,11
14	011	1	0	00	Miss confl			18,19	14,15
11	010	1	1	00	Miss confl			10,11	14,15
15	011	1	1	00	Hit			10,11	14,15

- Rechner C:

Anzahl Misses: 3

Adr	Adr binär			Hit/Miss	Ind 0			
	tag	owr	oby		Block 00	Block 01	Block 10	Block 11
19	1001	1	00	Miss comput	18,19			
10	0101	0	00	Miss comput	18,19	10,11		
19	1001	1	00	Hit	18,19	10,11		
14	0111	0	00	Miss comput	18,19	10,11	14,15	
11	0101	1	00	Hit	18,19	10,11	14,15	
15	0111	1	00	Hit	18,19	10,11	14,15	

(c) Betrachten Sie die folgenden Fehlzugriffsraten:

- Rechner A: Bei Instruktionen 4%, bei Daten 8%.
- Rechner B: Bei Instruktionen 2%, bei Daten 5%.
- Rechner C: Bei Instruktionen 2%, bei Daten 4%.

Bei den drei Rechnern seien  $\frac{3}{5}$  aller Instruktionen Speicherinstruktionen, die Miss-Strafe betrage 7 Zyklen bei Rechner A und 11 Zyklen bei den Rechnern B und C. Welcher Rechner wendet am meisten Zyklen für die Behandlung von Cache-Misses auf?

**Lösung:**

$$\text{Miss-Zyklen} = \text{Instruktions-Miss-Zyklen} + \text{Daten-Miss-Zyklen} \quad (1)$$

$$= \# \text{Instruktionen} \times \text{Instruktions-Miss-Rate} \times \text{Miss-Straf-Zyklen} \quad (2)$$

$$+ \frac{3}{5} \times \# \text{Instruktionen} \times \text{Daten-Miss-Rate} \times \text{Miss-Straf-Zyklen} \quad (3)$$

$$= \# \text{Instruktionen} \times \text{Miss-Straf-Zyklen} \quad (4)$$

$$\times (\text{Instruktions-Miss-Rate} + \frac{3}{5} \times \text{Daten-Miss-Rate}) \quad (5)$$

$$\frac{\text{Miss-Zyklen}}{\# \text{Instruktion}} = \text{Miss-Straf-Zyklen} \times (\text{Instruktions-Miss-Rate} + \frac{3}{5} \times \text{Daten-Miss-Rate}) \quad (6)$$

- Rechner A:  $\frac{\text{Miss-Zyklen}}{\# \text{Instruktion}} = 7 \cdot (0.04 + \frac{3}{5} \cdot 0.08) = 0.616$

- Rechner B:  $\frac{\text{Miss-Zyklen}}{\# \text{Instruktion}} = 11 \cdot (0.02 + \frac{3}{5} \cdot 0.05) = 0.55$

- Rechner C:  $\frac{\text{Miss-Zyklen}}{\# \text{Instruktion}} = 11 \cdot (0.02 + \frac{3}{5} \cdot 0.04) = 0.484$

Rechner A wendet also die grösste Anzahl von Zyklen für die Behandlung von Cache-Misses auf.

- (d) Wir möchten Rechner C um den Faktor 4 beschleunigen unter der Annahme, dass die Speicherzugriffszeiten unverändert bleiben. Um welchen Faktor muss dann die Taktrate erhöht werden? Der CPI bei perfektem Cache sei 3.388, bezüglich der Miss-Raten gelten die Angaben aus Teilaufgabe c.

**Lösung:**

Sei  $k$  der gesuchte Faktor,  $\text{CPUZeit}_{\text{normal}}$  die normale Rechenzeit von Rechner C und  $\text{CPUZeit}_{\text{schnell}}$  die Rechenzeit des beschleunigten Rechners. Da sich die Speicherzugriffszeiten nicht ändern, hat der beschleunigte Rechner  $k$ -mal so viele Miss-Zyklen wie Rechner C. Es gilt nun (siehe Kapitel 11, Seite 30 im Skript):

$$\begin{aligned} 4 &= \frac{\text{CPUZeit}_{\text{normal}}}{\text{CPUZeit}_{\text{schnell}}} = \frac{\text{CPI}_{\text{normal}} \times \text{Instruktionen} \times \frac{1}{\text{Taktrate}}}{\text{CPI}_{\text{schnell}} \times \text{Instruktionen} \times \frac{1}{k \times \text{Taktrate}}} \\ &= \frac{\text{CPI}_{\text{perfekt}} + 0.484}{\left(\text{CPI}_{\text{perfekt}} + k \times 0.484\right) \times \frac{1}{k}} = \frac{k \times 3.872}{3.388 + k \times 0.484} \cdot \\ &\Rightarrow k = 7 \end{aligned}$$