

Rechnerorganisation im WS 2020/21

Musterlösungen zum 09. Übungsblatt

Prof. Dr. Jörg Henkel
Dr.-Ing. Lars Bauer
Roman Lehmann, M. Sc.
Haid-und-Neu-Str. 7,
Geb. 07.21 (Technologiefabrik)
Email: roman.lehmann@kit.edu

Lösung 1

(8 Punkte)

1. Die Größe des Caches kann berechnet werden.

2 P.

Datenblock:

$$\text{Anzahl an Sätzen: } \frac{\text{Zeilen}}{\text{Assoziativität}} = \frac{32}{8} = 4$$

⇒ Satzindex: $4 = 2^2$, somit 2 Bit für den Satzindex

$$\Rightarrow \text{Offset: } 32 \text{ Bit} - 20 \text{ Bit} - 2 \text{ Bit} = 10 \text{ Bit}$$

$$\text{Größe einer Cache-Zeile: } 2^{10} \text{ Byte} = 1024 \text{ Byte}$$

Bei 32 Cache-Zeilen belegen die im Cache gespeicherten Daten dann 32 KByte ($K = 2^{10} = 1024$).

Gesamter Cache:

Um die Größe des gesamten Caches zu berechnen müssen pro Cache-Zeile noch die Breite des Tags und die Anzahl der Statusbits dazu gezählt werden.

$$\text{Das sind: } 32 * (20 \text{ Bit} + 2 \text{ Bit}) = 32 * 22 \text{ Bit} = 704 \text{ Bit} = 88 \text{ Byte}$$

$$\text{Gesamtgröße: } 32 \text{ KByte} + 88 \text{ Byte} = 32856 \text{ Byte}$$

2. Die Größe eines Cache-Blocks kann auf Basis der angegebenen Daten nicht berechnet werden.

2 P.

3. Die Assoziativität des Caches kann berechnet werden.

2 P.

$$\text{Offset bei einer Blockgröße von 64 Byte: } 64 = 2^6, \text{ somit 6 Bit}$$

$$\text{Die Bits des Satzindex ergeben sich zu: } 32 \text{ Bit} - 26 \text{ Bit} - 6 \text{ Bit} = 0 \text{ Bit}$$

Es gibt somit nur einen Satz und es handelt sich somit um einen vollassoziativen Cache.

Die Assoziativität entspricht damit der Anzahl an Cachezeilen. Da es $\frac{512 \text{ KByte}}{64 \text{ Byte}} = 8K$ ($K = 2^{10} = 1024$) Cachezeilen gibt, ist die Assoziativität also 8192.

4. Bei einem Cache Hit kann bei einem Cache mit Rückschreibeverfahren niemals ein Schreibzugriff ausgelöst werden, da nur Cache Misses zu einem Schreibzugriff führen können (falls das Dirty-Bit der verdrängten Cachezeile gesetzt ist).

2 P.

Daher liegt die obere Schranke für die Anzahl der tatsächlich durchgeführten schreibenden Speicherzugriffe bei 2.

Eine untere Schranke größer Null kann nicht angegeben werden.

Lösung 2

(12 Punkte)

• Unterteilungen

3 P.

- AV: 28 Bit Tag, 4 Bit Offset
- A2: 27 Bit Tag, 1 Bit Satzindex (SI), 4 Bit Offset
- DM: 26 Bit Tag, 2 Bit Zeilenindex (ZI), 4 Bit Offset

• Siehe Tabelle:

9 P.

Adresse	AV	A2	DM	SI	ZI
0x25	-	-	-	0	2
0x3A	-	-	-	1	3
0x12	-	-	-	1	1
0x74	-	-	-	1	3
0x36	X	-	-	1	3
0x08	-	-	-	0	0
0x09	X	X	X	0	0
0x16	X	-	X	1	1
0x28	-	X	X	0	2
0x52	-	-	-	1	1
0x22	X	X	X	0	2
0x11	X	X	-	1	1

Lösung 3

(6 Punkte)

1. Es werden je 333 Lade- und Speicherbefehle ausgeführt, d.h. insgesamt 666 Befehle, die potentiell auf den Speicher zugreifen.

2 P.

Bei der Breite einer Cachezeile von 16 Byte erzeugt das Programmstück $\lceil \frac{334}{4} \rceil = 84$ Compulsory Misses beim Zugriff auf das (im Speicher an einem Vielfachen von 16 ausgerichteten) Array. Es liegen somit $666 - 84 = 582$ Cache Hits vor.

2. Die Hit-Rate beträgt $\frac{582}{666} = 87,387\%$.

2 P.

Die mittlere Zugriffszeit beträgt $0,87387 \cdot 2 + (1 - 0,87387) \cdot 10 = 3,009$ Taktzyklen.

3. Zwei Cachezeilen genügen.

2 P.

Eine Zeile genügt nicht, da sonst alle vier Arrayelemente zusätzlich zu dem einen Compulsory Miss zwei Capacity Misses auftreten.

Lösung 4

(2 Punkte)

	wahr	falsch
Bei einem direktabgebildeten Cache ist es nicht notwendig, eine Block-Ersetzungsstrategie festzulegen.	×	
Die Verwendung des Rückschreibverfahrens (<i>write back policy</i>) verhindert das Auftreten von Konsistenzproblemen bei Mehrprozessorsystemen.		×
Bei einem satzassoziativen Cache können mehrere Cachezeilen gültig sein und denselben Tag, aber dennoch unterschiedliche Daten enthalten. Bei einem vollassoziativen Cache kann das nicht passieren.	×	
Je höher die Assoziativität eines Cache, desto weniger Komparatoren werden zu seiner Realisierung benötigt. Allerdings steigt damit auch die Auftrittswahrscheinlichkeit für Conflict Misses.		×